VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

# WEBOVÁ APLIKACE VYUŽÍVAJÍCÍ VÍCEFAKTOROVOU AUTENTIZACI

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

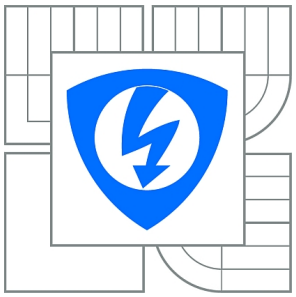
AUTOR PRÁCE
AUTHOR

Bc. JAN HUMPOLÍK

BRNO 2013


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY


FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS


# WEBOVÁ APLIKACE VYUŽÍVAJÍCÍ VÍCEFAKTOROVOU AUTENTIZACI

WEB APPLICATION UTILIZING MULTI-FACTOR AUTHENTICATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS


AUTOR PRÁCE — Bc. JAN HUMPOLÍK

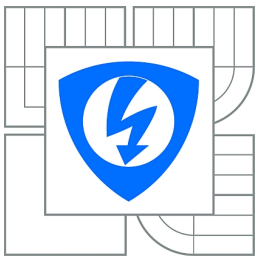
AUTHOR

VEDOUCÍ PRÁCE — Ing. RADEK DOLEŽEL

SUPERVISOR



BRNO 2013



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky a komunikačních technologií**

**Ústav telekomunikací**

# Diplomová práce

magisterský navazující studijní obor
**Telekomunikační a informační technika**

***Student:*** Bc. Jan Humpolík ***ID:*** 106483
***Ročník:*** 2 ***Akademický rok:*** 2012/2013

**NÁZEV TÉMATU:**

## Webová aplikace využívající vícefaktorovou autentizaci

**POKYNY PRO VYPRACOVÁNÍ:**

Prostudujte současné trendy zabývající se vícefaktorovou autentizací s využitím hardwarových prostředků a zaměřte se na možnosti implementace ve webovém prostředí. Navrhněte webovou aplikaci, která bude pro zpřístupnění svého obsahu využívat vícefaktorovou autentizaci realizovanou pomocí kryptografických čipových karet a chytrých mobilních telefonů. Návrh programově implementujte a proveďte celkové zhodnocení bezpečnosti.

**DOPORUČENÁ LITERATURA:**

[1] DOSTÁLEK, Libor, Marta VOHNOUTOVÁ a Miroslav KNOTEK. Velký průvodce infrastrukturou PKI a technologií elektronického podpisu. 2., aktualiz. vyd. Brno: Computer Press, 2009, 542 s. ISBN 978-80-251-2619-6.

[2] MENEZES, Alfred J, Paul C OORSCHOT a Scott A VANSTONE. Handbook of applied cryptography. Vyd. 1. Boca Raton: CRC Press, 1997, 780 s. ISBN 0-8493-8523-7.

***Termín zadání:*** 11.2.2013 ***Termín odevzdání:*** 29.5.2013

***Vedoucí práce:*** Ing. Radek Doležel
***Konzultanti diplomové práce:***

**prof. Ing. Kamil Vrba, CSc.**
*Předseda oborové rady*

**UPOZORNĚNÍ:**

## ABSTRAKT

V práci je popsáno a implementováno 5 autentizačních metod (některé s vlastním návrhem) vícefaktorové autentizace v prostředí webových aplikací. Z výsledků práce lze využít jak webovou aplikaci, tak jednotlivé autentizační metody (jenž jsou přiloženy samostatně) pro použití ve vlastní webové aplikaci.

## KLÍČOVÁ SLOVA

Android, autentizace, certifikát, čipová karta, Google Authenticator, MyWebID, OTP, SMS autentizace, vícefaktorová autentizace, webový server, YubiKey

## ABSTRACT

In the thesis are described and implemented 5 methods (some with their own proposal) of multifactor authentication in web application environment. The results of the work is the web application and individual authentication methods (which are attached separately) for use in your own web application.

## KEYWORDS

Android, authentication, Google Authenticator, multi-factor authentication, MyWebID, OTP, smart card, SMS authentication, USB token, web server, YubiKey

HUMPOLÍK, J. *Webová aplikace využívající vícefaktorovou autentizaci*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 51 s. Vedoucí práce Ing. Radek Doležel

# PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Webová aplikace využívající vícefaktorovou autentizaci“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ..... .....

(podpis autora)

# PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Radku Doleželovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.


V Brně dne ..... .....

(podpis autora)

# OBSAH

# SEZNAM OBRÁZKŮ

# SEZNAM TABULEK

# ÚVOD

Na úvod je třeba říci, že tato diplomová práce se zabývá teorií jen do určité míry, tak aby byly získány potřebné teoretické základy, které poskytnou potřebný odrazový můstek pro další empirický výzkum. Pokud možno se nezabývá detailním rozborem, který se dá získat přes odkazy z citací. Výsledkem práce jsou jednak autentizační metody, a také webová aplikace MyWeb, která obsahuje i nápady na UI/UX (User Interface/User Experience) těchto autentizačních metod s uživateli, včetně uživatelova vlastního nastavení, logování přístupů a změn, blokování přístupů z IP adres aj.

Práce může posloužit jako základ ke studiu a doplnění aplikace o další autentizační faktory, případně může čtenář jednoduše využít tyto autentizační metody (podle jednotlivých ukázek v příloze) a implementovat je ve své vlastní aplikaci.

## Co je to vícefaktorová autentizace

Autentizace (ověření identity uživatele důvěryhodnou autoritou) může být trojího typu: uživatel „něco ví“ (heslo), „něco má“ (čipová karta) nebo něco je (biometrika). Kombinací těchto typů vzniká vícefaktorová autentizace, v této práci se budeme věnovat kombinaci „něco ví“ a „něco má“.

## Současný stav vícefaktorová autentizace na webu

Google účet lze zabezpečit využíváním TOTP autentizace přes Google Authenticator, ve vybraných zemích poskytuje i SMS autentizaci a uvažuje se [1] o nasazení systému YubiKey.

Twitter po letech očekávání, některé metody vícefaktorové autentizace, momentálně nasazuje autentizaci formou SMS, opět ale jen ve vybraných zemích. Bohužel se zásadnější chybou návrhu [2], kdy útočník může vypnout dodatečné zabezpečení tak, že pošle SMS s podvrženým číslem s požadavkem STOP. Dokonce lze provést i opak, nastavit jinému uživateli SMS autentizaci a tím mu znemožnit přihlášení ke svému účtu.

Další významní hráči, jako Facebook, nebo internetové bankovnictví místních bank nabízí především SMS autentizaci. Jak se ale budete moci v této práci dočíst, není to dnes již zrovna nejvhodnější způsob, ačkoliv je stále široce používán.

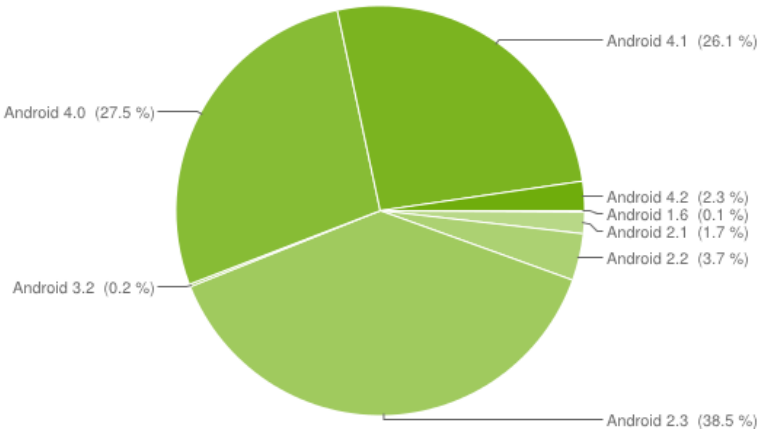
# 1 DRUHÝ AUTENTIZAČNÍ FAKTOR

## 1.1 MyWebID

MyWebID je v DP vytvořená aplikace pro autentizaci zařízením pro Android [1], (např. mobilním telefonem, tabletem) jako prostředku vícefaktorové autentizace.

Jednorázové autentizační kódy se zasílají skrze internetové připojení ze serveru na zařízení, následně je nutný ještě manuální opis kódu uživatelem do webové aplikace (viz obr. 1.2). Tato varianta sice vyžaduje internetové připojení, ale na druhou stranu není náchylná na únik algoritmů a inicializačních kódů jako u TOTP (Time-based One-time Password Algorithm) či HOTP (HMAC-based One Time Password) řešení, a dále je algoritmus o poznání snazší (vyvarování se bezpečnostní chybě v algoritmu).

Aplikace vyžaduje jako běhové prostředí minimálně Android 2.2 Froyo (dostupný přibližně od poloviny roku 2010) a nainstalovaný Google Play (není nutné distribuovat aplikaci skrze něj, ale je potřebný pro funkčnost GCM).

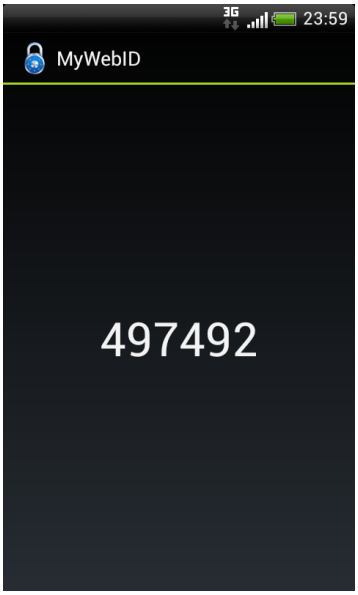



Obr. 1.1: Zastoupení verzí systému Android na zařízeních s přístupem na Google Play. Android 2.2 a vyšší má 98,3 % zařízení. Data [3] platná k 1.5.2013.

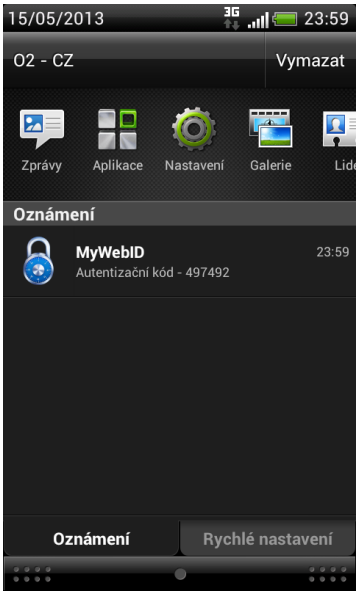
### 1.1.1 Dotazování vs oznámení

Mnoho aplikací pro mobilní telefony vyžaduje data z internetu. Jedním ze způsobů pro aktualizaci těchto dat je periodické dotazování serveru na nová data, ale když

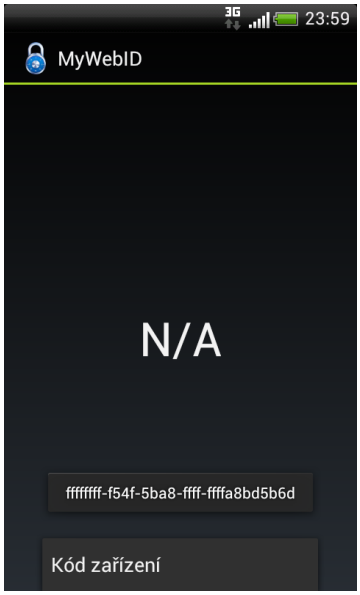
[1] Android je operační systém společnosti Google Inc., který naléza využití v mobilních telefonech, tabletech a různých specifických zařízení spotřební elektroniky. Systém byl představen ve své první verzi 23. září 2008, v březnu 2013 měl již přes 750 miliónu [4] aktivací a denně přibývá přes 1,5 miliónu nových [5]. Očekává se [4], že koncem roku 2013 Android dosáhne 1 miliardy aktivací.




Obr. 1.2: MyWebID.




Obr. 1.3: Notifikace.




Obr. 1.4: Registrace.

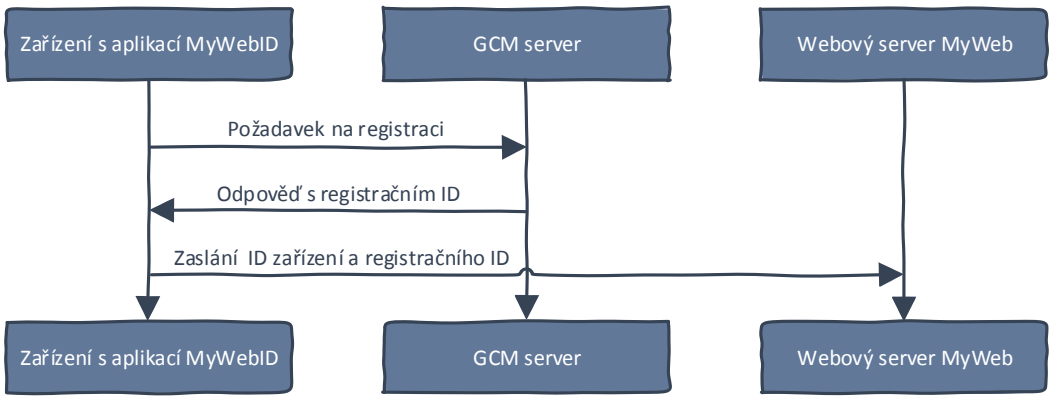
neexistují nová data, vede tato metoda ke zbytečným ztrátám na internetové přípojce (FUP, propustnost) a akumulátoru mobilního telefonu. Alternativou je druhá možnost, kdy server kontaktuje mobilní aplikaci jen v případě, že jsou nová data k dispozici (push notifikace). Jestliže se data nemění v pravidelných intervalech, může být tato metoda výhodnější, ale jak si ukážeme dále, vyžaduje specifickou infrastrukturu navíc.

### 1.1.2 Google Cloud Messaging

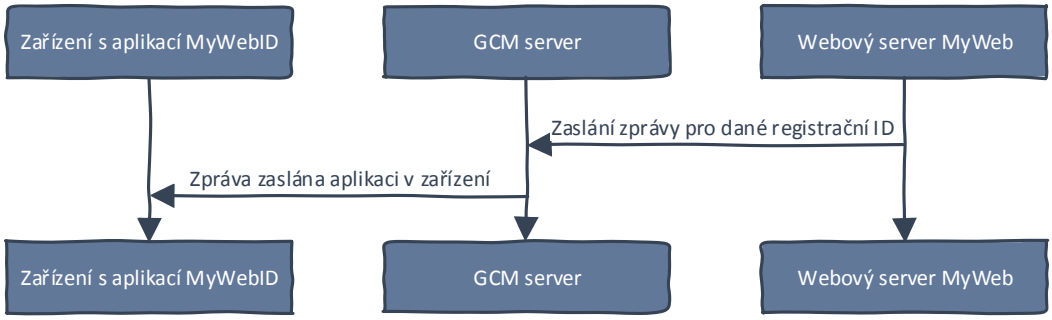
Od Androidu 2.2 jsou k dispozici push notifikace i pro veřejnost (dříve jen pro Google aplikace), tato služba se jmenuje GCM (Google Cloud Messaging) [6].

Když chce aplikace na serveru zaslat data do Android aplikace, zašle zprávu skrze POST metodu na GCM server, a ten zprávu přepošle do zařízení (mobilní telefon, tablet atp.). Jestliže není zařízení dostupné, zpráva bude zaslána znovu, až opět bude. Jakmile je zpráva přijata zařízením, je vytvořena notifikace, jejíž otevřením se otevře daná aplikace, která už svým způsobem přijatá data dále zpracuje.

GCM zprávy jsou omezeny na velikost 4 KB, jsou navrženy pro informování o dostupnosti nových dat, ne pro jejich přenos. Typická praxe je, že GCM server oznámí Androidí aplikaci, že jsou dostupná nová data, a následně si aplikace stáhne data z jiného serveru. GCM využívá existující spojení s Google servery pro minimalizaci datových přenosů a spotřeby elektrické energie akumulátoru.




Obr. 1.5: Jednorázová registrace zařízení pro použití GCM.



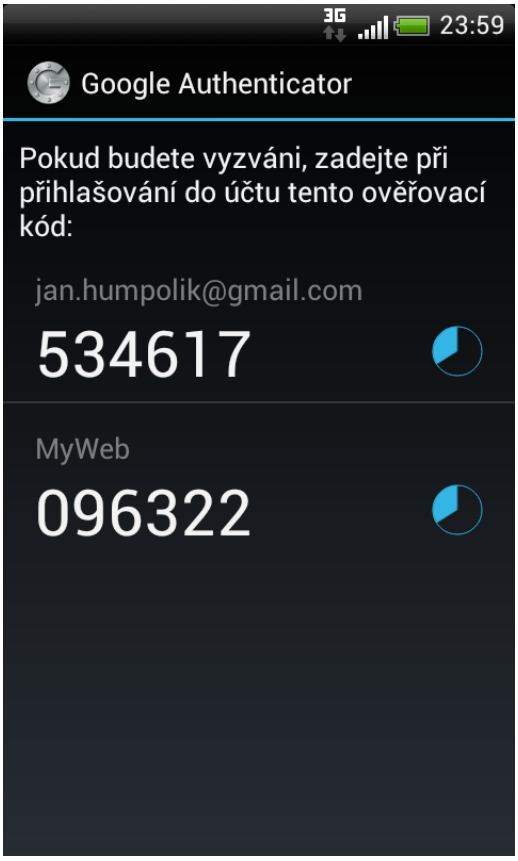

Obr. 1.6: Způsob následného zasílání zpráv.

## 1.2 Google Authenticator

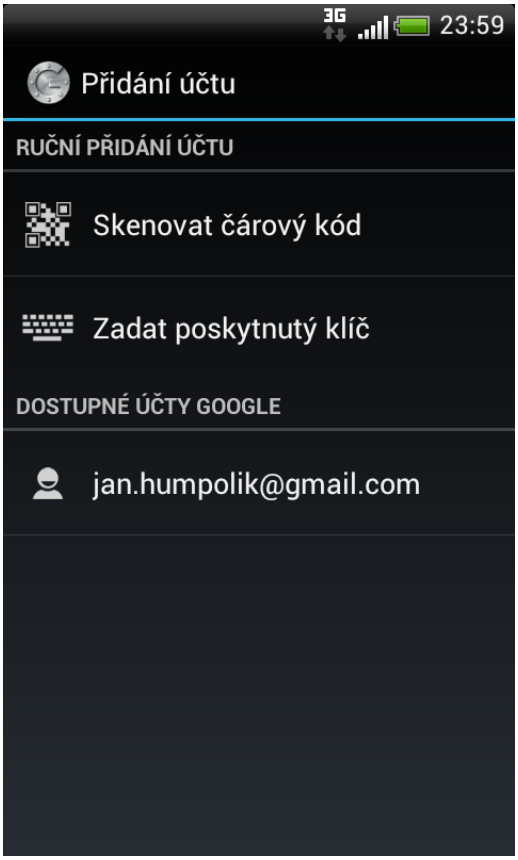
Google Authenticator je aplikace pro Android, BlackBerry nebo iPhone pro generování jednorázových kódů přímo na těchto zařízeních.

Je založen na RFC 4226 – Time-based One-time Password Algorithm (TOTP), který se inicializuje pomocí 16místného `base32` kódu (RFC 4648). Google také nabízí PAM (Pluggable Authentication Module) umožňující uživatelům integrovat vícefaktorovou autentizaci do `sshd` (OpenSSH Daemon). [7]

Modul může být napsán, na podporu Google TOTP, v jakémkoli jazyce – Jedinou námitkou s psaním knihovny pro PHP je chybějící podpora `base32` funkcí, jenž vyžaduje RFC 4648. To je pravděpodobně nejvíce choulostivá část implementace pro PHP, ale to dostupné knihovny již mají vyřešené [8].




Obr. 1.7: Google Authenticator.




Obr. 1.8: Přidání účtu.

## 1.3 SMS autentizace

Tato metoda umožňuje realizovat autentizaci na základě zasílání časově omezených jednorázových kódů pomocí SMS zpráv. Použití této autentizační metody nevyžaduje na klientské straně žádné specializované zařízení a SMS autentizace tak představuje široce dostupnou autentizační metodu. Při použití tzv. chytrých telefonu je ale mj. náchylná na *malware*.

Cena SMS nemusí být nijak vysoká, v případě využití SMS brány od společnosti Hosting90 [9] lze při zakoupení kreditu v hodnotě 5 000 Kč dosáhnout ceny 0,25 Kč vč. DPH. Služba je dostupná jen v České republice. V případě že by bylo potřeba zasílat SMS mezinárodně, lze využít službu Twilio SMS [10].

## 1.4 YubiKey

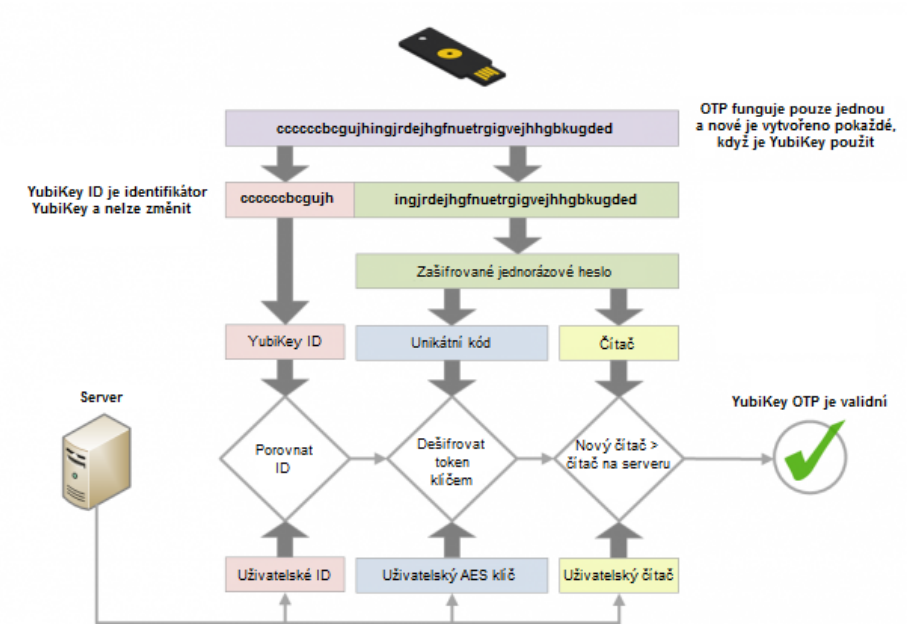
YubiKey je HW token vzhledově podobný klasickým bezpečnostním tokenům, ale postavený na úplně jiném principu. To má své nevýhody, zejména co se bezpečnosti

týče, ale také nezanedbatelnou výhodu vysoké kompatibility bez potřeby speciálních ovladačů. YubiKey se vůči počítači tváří jako běžná USB klávesnice, která na vyžádání vyšle určitou sekvenci znaků. Na klasickém YubiKey je „tlačítko“ (optický senzor), které rozlišuje krátký (0,3 až 1,5 sekundy) a dlouhý stisk (2,5 až 5 sekund), lze tak generovat jednorázové heslo typu HOTP, resp. statické heslo (16 až 64 znaků) [11]. Neobsahuje baterii ani žádné pohyblivé části. Existují i další varianty (obr. 1.9), YubiKey NEO [12] (klasický YubiKey s podporou NFC, díky němuž lze využít YubiKey NEO např. u chytrých mobilních telefonů) a YubiKey Nano [13] (miniaturní YubiKey pro stálé umístění v USB portu, zmáčknutím zlatavého okraje v USB portu dojde k opětovnému generování hesla).

Obr. 1.9: Varianty klasického YubiKey, YubiKey NEO a YubiKey Nano [14].

### 1.4.1 Režim jednorázového hesla

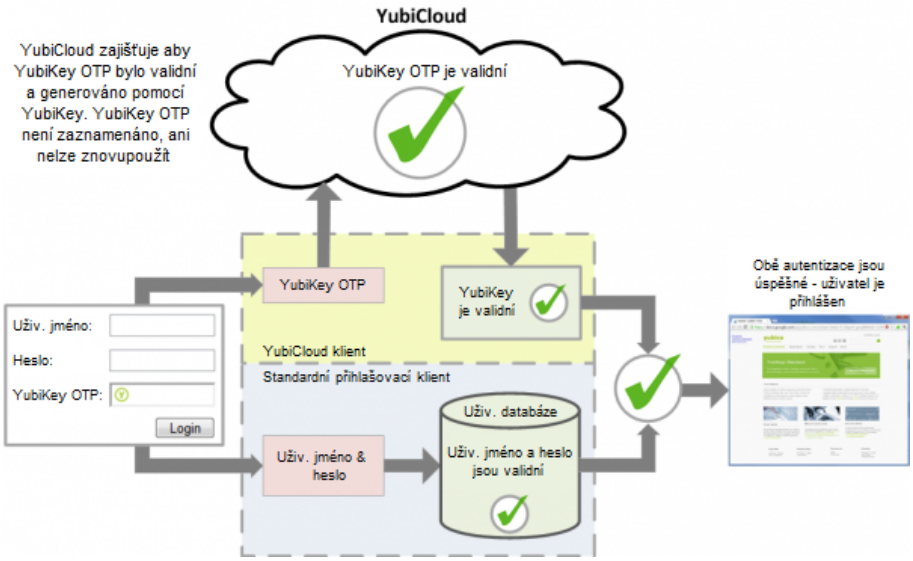
Při stisku „tlačítka“ YubiKey poskládá balíček dat složený z privátního ID (řetězec znaků stanovený uživatelem, jinak nuly), čítače stisků tlačítka, dalšího čítače stisků tlačítka jen v dané relaci, časové značky, náhodného čísla a kontrolního součtu toho všeho. Tento balíček zašifruje AES šifrou se 128bitovým klíčem (také stanoveným uživatelem) a zakóduje do sekvence kláves, kterou pošle počítači. Autentizační program převezme tuto sekvenci, odšifruje AES (protože mu uživatel už dříve řekl šifrovací klíč) a zkontroluje CRC (a případně privátní ID. Tím si ověří, že řetězec přišel ze správného YubiKey. Následně se podívá na čítač stisků a porovná si ho se svojí uloženou hodnotou. Pokud je čítač menší nebo roven, jde o opakovaný klíč a autentizační program ho odmítne. V opačném případě si uloží načtenou hodnotu čítače a přihlášení uživatele povolí.) [15, 16]. Zjednodušené schéma viz obr. 1.10.




Obr. 1.10: YubiKey – režim jednorázového hesla [17].

### 1.4.2 YubiCloud

Jeden profil YubiKey se ale dá použít jen na jednom serveru (klidně i ve více aplikacích, které ovšem budou používat společnou databázi čítačů), jinak padá jednorázovost (pokud nejsou databáze propojené, tak se první server nedozví o použití YubiKey na serveru druhém, a „jednorázové“ heslo z druhého serveru půjde použít i na něm) [15].




Obr. 1.11: YubiCloud – princip použití [18].

I z tohoto důvodu lze využít YubiCloud, díky němuž bude jediná autentizační databáze v *cloudu*. Navíc nebude nutné nové YubiKey tokeny nijak přeprogramovat a uživatelé si tak mohou sami YubiKey zakoupit (viz [19]) a nastavit jeho používání ve webové aplikaci. Je tak z důvodu, že YubiKey má již z výroby naprogramované individuální nastavení pro YubiCloud. V případě, že uživatel následně YubiKey přeprogramuje [20] (např. z důvodu nastavení statické hesla, jakožto druhého módu), tak se stane pro YubiCloud nefunkční. Lze jej ale pro YubiCloud opět nastavit zasláním nových údajů do YubiCloudu, viz [21].

**Postup pro implementaci**

1. Získáme YubiKey, např. z oficiální e-shopu [19].
2. Vygenerujte Client ID + API key na [22] (je nutné již mít YubiKey).
3. Stáhneme knihovnu pro práci s YubiCloud [23], vyžadující podporu `curl` (`mod_curl` v Apache) a PHP knihovnu PEAR [24]. Případně lze využít tyto knihovny předpřipravené na CD, v tomto případě již nejsou nutné další kroky.
4. Implementujeme, např. podle následujícího příkladu.
5. Doporučuji potlačit PHP chybové hlášení `E_STRICT`, pomocí `error_reporting (E_ALL & ~E_NOTICE & ~E_STRICT);` Z důvodu, že práce mezi PEAR knihovnou a YubiCloud knihovnou již není podle nejnovějších standardů PHP, a proto se objevují varovná hlášení.

## 1.5 Certifikát (čipová karta)

Tato část se věnuje přihlašování pomocí certifikátů a k nim asociovaným privátním klíčům, které jsou uloženy právě na čipových kartách. Karty neobsahují uživatelská hesla. Tajná informace je reprezentována právě privátním klíčem, který nikdy neopouští kartu a karta je tak nekopírovatelná. Karty obsahují kryptografický čip, svůj vlastní operační systém a bezpečnou paměť na privátní klíče. To znamená, že mohou veškeré kryptografické operace provádět mimo operační systém počítače, ke kterému jsou připojeny. Jestliže je v operačním systému například virus, má jen velmi omezené možnosti, jak se dostat k obsahu karty a jak jej zneužít. [25]

**Podmínky implementace**

1. Přístup ke konfiguračním souborům Apache.
2. Nutná podmínka pro autentizaci osobním certifikátem je mít web zabezpečený pomocí TLS/SSL.
3. Upravit nastavení HTTPS VirtualHostu v Apache, ke stávajícímu funkčnímu nastavení SSL přidat:

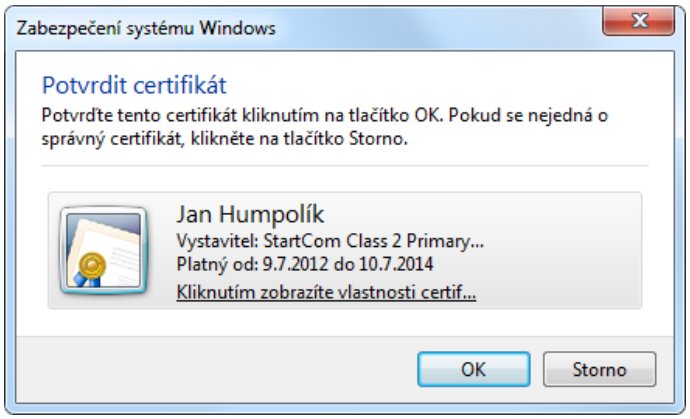
`SSLVerifyClient require`
*zapnutí ověření certifikátem*
`SSLVerifyDepth 2`
*kolik úrovní ještě strpíme (0 = self-signed, 1 = 1 podřízená CA, 2 = 2 ...)*
`SSLCACertificateFile "C:/Web/system/Apache/conf/certs/ca.pem"`
*CA vůči které ověřujeme certifikáty*
`SSLOptions +StdEnvVars`
*export do systémových proměnných (např. pro PHP)*

### Jednoduchá ukázka v PHP

```
if ($_SERVER['SSL_CLIENT_VERIFY'] == "SUCCESS" &&
    $_SERVER['SSL_CLIENT_S_DN_Email'] == "jan.novak@example.com") {
  echo "Úspěšná autentizace";
} else {
  echo "Certifikát platný, ale neodpovídá e-mailové adrese";
}
```

### Poznámky

- V případě, že certifikát není platný, nebo dojde uživatelem k odmítnutí, tak Apache sám zamítne přístup, proto pro tuto situaci není v PHP podmínka.
- V osobním certifikátu musí být přítomna e-mailová adresa daného subjektu.
- Protože je ověřována platnost osobního certifikátu, musíme zadat cestu k certifikátu CA, z toho plyne závislost na některé CA – např. StartSSL (obr. 1.12).
- Neověří-li se platnost osobního certifikátu, pak autentizace není dostatečná, může tak kdokoli s certifikátem a daným e-mailem být autentizován.




Obr. 1.12: Dialogové okno z předchozího příkladu, certifikát od CA StartSSL.

## 1.6 Srovnání použitých metod

| Autentizační metoda | Výhody | Nevýhody | Pořízení | Provoz |
|---|---|---|---|---|
| MyWebID | Postačí libovolné Android zařízení s přístupem na internet. | Závislost na Google (jak na Androidu, tak na GCM). Nutný přístup k internetu. Riziko malware. | od 0 Kč | od 0 Kč (připojení zařízení k internetu) |
| Google Authenticator | Podpora více tokenů na jednom zařízení. Multiplatformní. | Riziko rozsynchronizace (TOTP). Riziko malware. | od 0 Kč | 0 Kč |
| SMS autentizace | Mobilní telefon s podporou SMS má téměř každý. | Občasné prodlevy v doručení. Nutné být na území pokrytým signálem mobilního operátora. U chytrých telefonů riziko malware. | od 0 Kč | od 0,25 Kč/SMS při zakoupení kreditu v hodnotě min. 5000 Kč, jinak od 0,50 Kč/SMS |
| YubiKey | Nevyžaduje žádné ovladače. Lze připojit na svazek klíčů. | Bez využití YubiCloudu je kvůli HOTP prakticky nutné mít ke každé službě jiný YubiKey. | od 330 Kč při koupi 50 ks; od 615 Kč při koupi 1 ks | 0 Kč |
| čipová karta | Nelze zkopírovat. Možnost využití Single Sign-On (SSO). | Instalace SW na klientský počítač, a tak i potřebná práva v OS. Bezpečností problémy JRE. Nejvyšší cena. | vč. čtečky karet až tisíce Kč | Obnova certifikátu. |

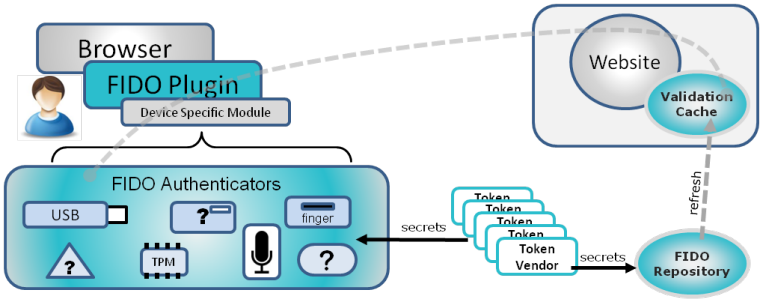
# 2 DALŠÍ WEBOVÉ AUTENTIZAČNÍ METODY

## 2.1 FIDO (Fast IDentity Online)

FIDO Alliance si dala za cíl sestavit standard pro přihlašování pomocí různých prostředků, které jsou bezpečnější než klasické heslo. Zatím jen na papíře. [26]

Budou tam patřit různé autentizační USB a NFC klíče, čipové karty, integrované TPM čipy, přihlašování pomocí otisku prstu, rozpoznání tváře, hlasu a tak dále. Všechny tyto systémy dnes, v proprietární podobě, již existují, chybí ale jednotné rozhraní. FIDO má docílit toho, že ať už se použije jakýkoliv z těchto prostředků, webová služba mu bude rozumět. [27, 26]

Nový protokol se označuje jako OSTP (Online Security Transaction Protocol) a jeho specifikace bude zveřejněna v druhé polovině roku 2013. Vývoj aplikace zajistí Nok Nok Labs – start-upová firma s investicí 15 milionů USD. Kromě Nok Nok Labs stojí za Aliancí FIDO řada významných firem, mezi zakladatele patří také PayPal, Lenovo, Validity Sensors, Agnitio a Infineon. Aliance FIDO byla zřízena jako nezisková organizace, takže se očekává, že nový protokol bude veřejný. [28]




Obr. 2.1: FIDO [27] bude kombinovat HW, SW a internetové služby.

## 2.2 mojeID

Služba mojeID [29] od sdružení CZ.NIC, které je správcem české národní domény, v současné době nabízí možnost vícefaktorové autentizace formou certifikátu, nebo TOTP (např. Google Authenticator).

Nepracuje s anonymními identitami, ale jen s identitami ověřenými. Validace probíhá formou opsání kódu z obdržené SMS a dopisu na zadanou adresu, případně zasláním úředně ověřeného formuláře do CZ.NIC.

Pro koncové uživatele je použití zdarma, pro implementátory je roční poplatek 1000 Kč. Existuje však Motivační program, kde lze za přivedené nové uživatele získat odměnu.

# 3 SERVEROVÁ ČÁST APLIKACE (MYWEB)

Navržená a vyvinutá webová aplikace MyWeb sdružuje (ve které jsou implementovány) všechny popisované autentizační metody z kapitoly 1. Princip autentizace je založen na myšlence liberalismu a minimalismu, kdy uživatel nemá od začátku pevně nastaven některý typ vícefaktorové autentizace, ale vývojář/administrátor nastaví každému modulu požadavek na autentizaci (jen heslo/vícefaktorová). Když poté uživatel otevře daný modul a zároveň bude autentizován do MyWeb pouze heslem, tak mu do daného modulu bude odepřen přístup a bude vyzván aby si vícefaktorovou autentizaci nastavil (obr. 3.2). Vícefaktorovou autentizaci si může kdykoliv změnit i zrušit, při zrušení po něm ale bude u daných modulů opět vyžadováno nastavení.

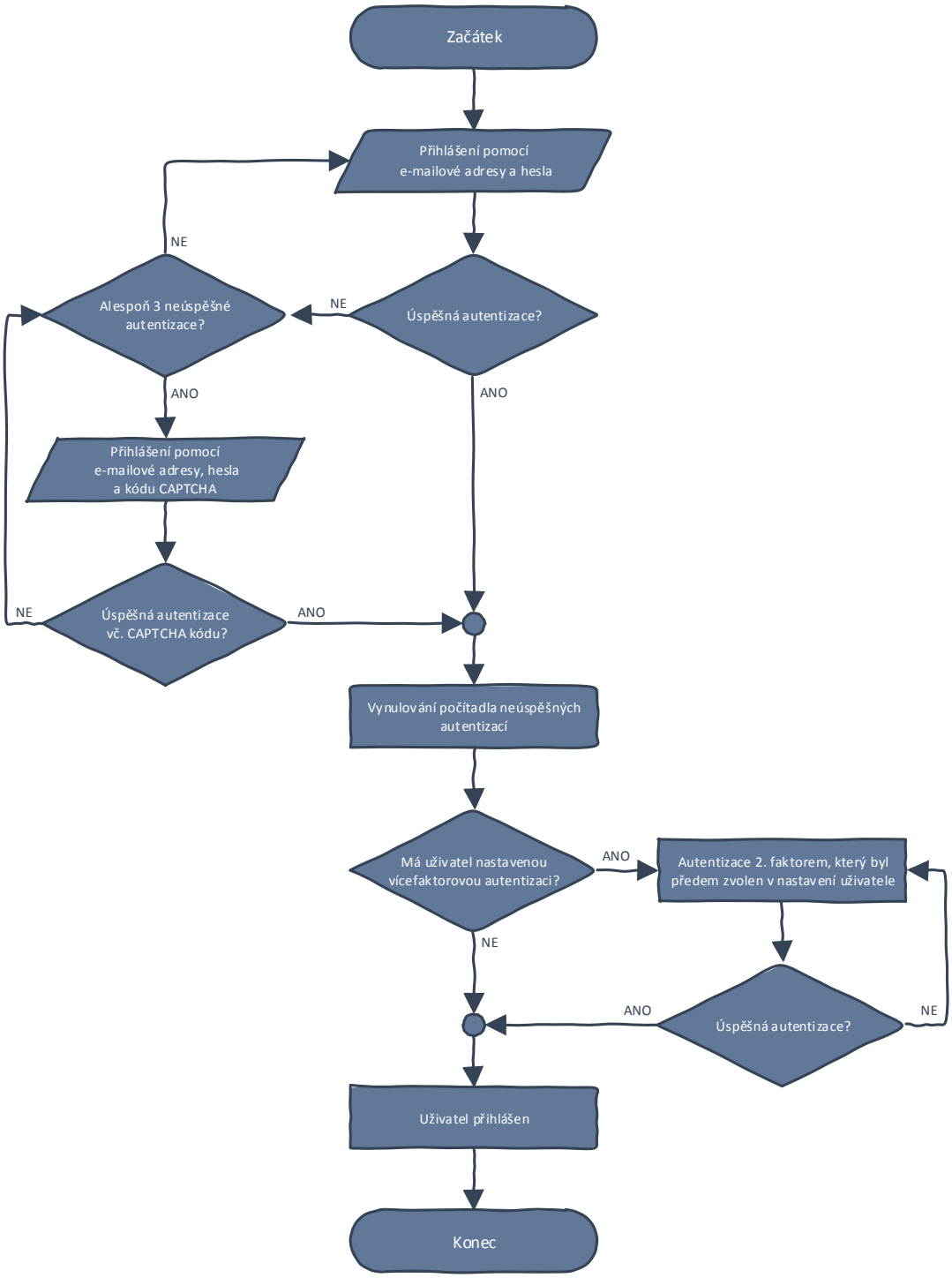
Přes snahu vytvořit tuto webovou aplikaci tak, aby byla co nejméně závislá na konfiguraci webového serveru, je nutné splnit alespoň následující podmínky: Apache 2 s funkčním HTTPS, podporou *mod_rewrite* a souborů *.htaccess*; PHP ve verzi 5.3 s rozšířeními *php_curl.dll*, *php_gd2.dll*, *php_mbstring.dll*, *php_mysqli.dll*, *php_openssl.dll*; MariaDB/MySQL 5 se zapnutým úložištěm InnoDB.

## 3.1 Šifrované spojení

Přihlašovací stránka webové aplikace automaticky přesměrovává na šifrované HTTPS spojení (v Chrome TLS 1.1, ostatní TLS 1.0/SSLv3), použitý serverový certifikát s veřejným klíčem RSA o délce 2048 bitů je podepsaný veřejně důvěryhodnou autoritou StartSSL [30] od společnosti StartCom Ltd., který je ve variantě „Class 1“ [31] dostupný na roční období zdarma. Zde je použit „Class 2“ – pro podporu *wildcard*.

Webový server Apache je nakonfigurován aby odolal BEAST útoku (Browser Exploit Against SSL/TLS) [32, 33], dokud se nepřejde na TLS 1.1 a vyšší (nutná podpora na web serverech i ve webových prohlížečích). Článek popisující tuto dočasnou ochranu, využívající proudovou šifru RC4 pro TLS 1.0 klienty a novější blokové šifry (AES, CAMELLIA, 3DES) pro klienty s podporou TLS 1.2, lze nalézt na [34]. Komplexně zkontrolovat TLS/SSL zabezpečení webového serveru lze otestovat na webu SSL Labs [35].

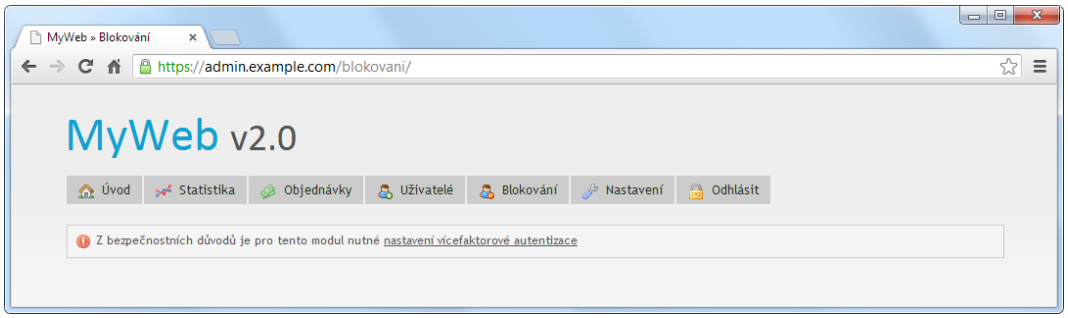
```
SSLProtocol all -SSLv2
SSLHonorCipherOrder On
SSLCipherSuite ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:RC4 \
   :HIGH:!MD5:!aNULL:!EDH
```




Obr. 3.1: Vývojový diagram autentizačního procesu aplikace MyWeb.

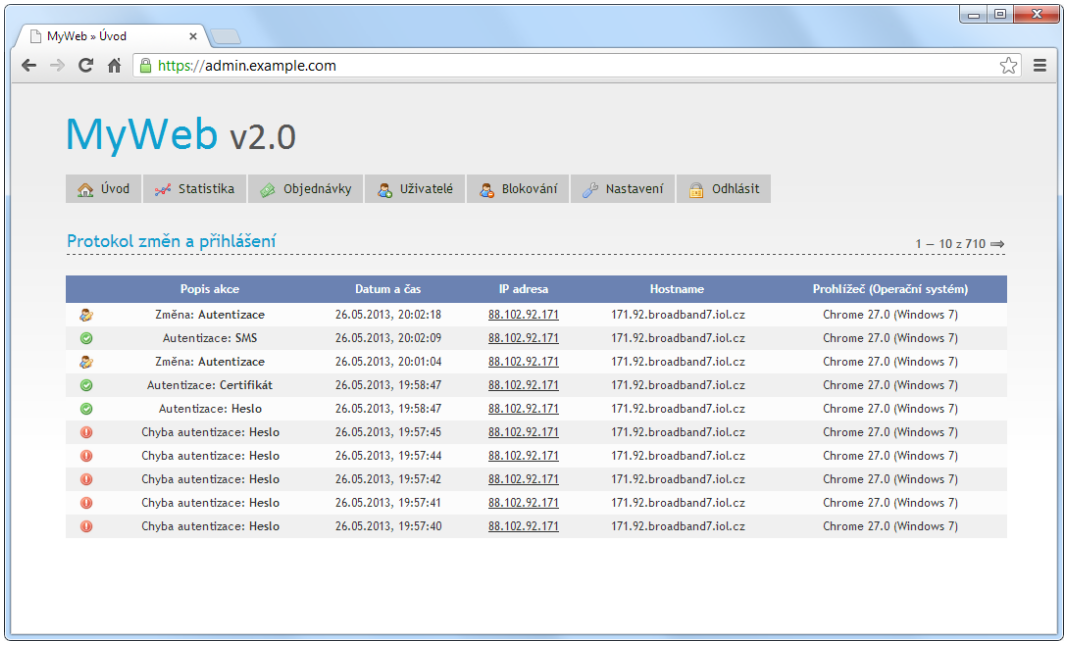
## 3.2 Bezpečnostní vlastnosti

– Hesla jsou uložena jako SHA-512 hashe (128 hexadecimálních znaků) s pseudonáhodně generovanou solí (ve tvaru md5 hashe) pro každého uživatele, která se vygeneruje nová při každé změně hesla. Jedná se tudíž o vysokou ochranu proti Rainbow tables, protože při změně hesla budou pracně vygenerované obrovské tabulky (512 bitů na pouhý 1 záznam bez režijních dat) k ničemu. Vedlejším efektem tohoto způsobu ukládání hesel je, že vytváří unikátní hash i pro případná stejná hesla více uživatelů.




Obr. 3.2: Modul vyžadující vícefaktorovou autentizaci.

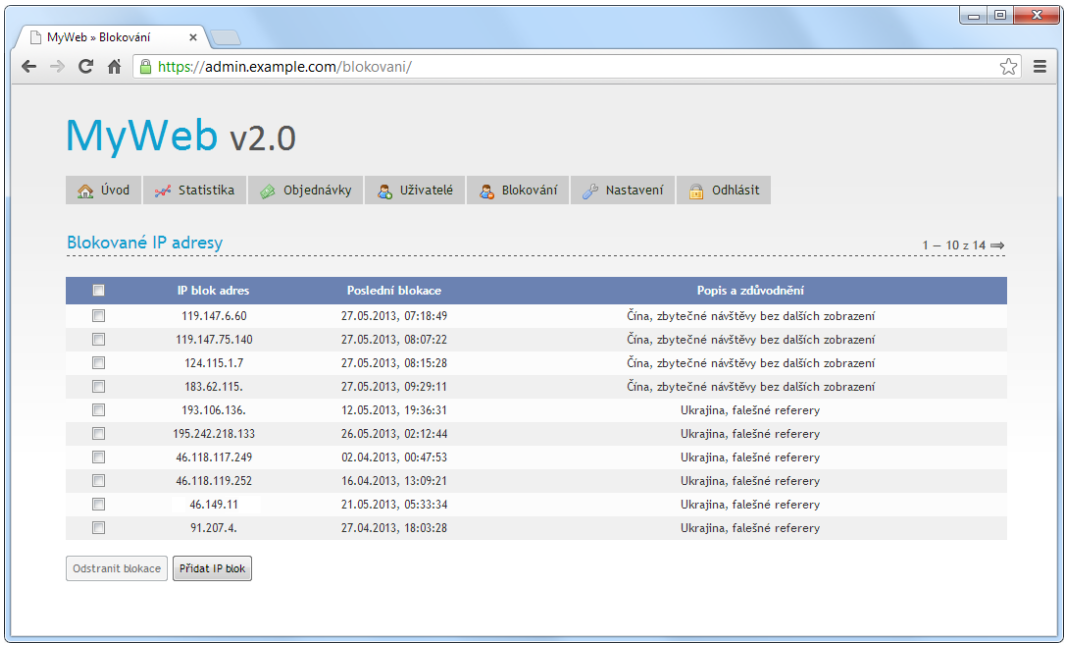
– Po úspěšném či neúspěšném přihlášení, změně hesla, nastavení atp. aplikace zaznamená IP adresu uživatele (podpora vč. IPv6), reverzní záznam, identifikátor jeho webového prohlížeče a čas přihlášení (obr. 3.3).




Obr. 3.3: Protokol změn a přihlášení po přihlášení.

– V případě 3 neúspěšných pokusů o přihlášení se spustí ochrana CAPTCHA (obr. 3.6), jejíž výstup uživatel zapisuje dohromady s heslem do stejné kolonky. Po úspěšném přihlášení je počet pokusů vynulován. Nízký počet možných pokusů je zvolen pro zastrašující efekt při „ručním“ zkoušení, při užití robota nemá nízký počet smysl. Počet neúspěšných pokusů se zaznamenává do databáze k přihlašovanému uživateli, nejsou tady závislé na cookies uživatele a není tedy možné získat další pokusy ani změnou celého počítače, natož vymazáním cookies.
– Zakázáno předávání session ID v URL, předávání je možné jen pomocí časově omezených cookies, které se neukládají na pevný disk a při znovuotevření webového prohlížeče je uživatel nucen se znovu přihlásit.
– Po úspěšném přihlášení je Session ID změněno – ochrana před Session Fixation.
– Skryta přítomnost PHP na serveru – v hlavičkách odpovědí neposílá informaci o PHP, jen zmínku o použití HTTP serveru Apache (bez verze). Aplikace neodkazuje na soubory s koncovkou .php na stejné doméně a nepoužívá výchozí session identifikátor PHPSESSID (používá „ID“).
– Ochrana proti XSS na úrovni převádění speciálních znaků na HTML entity také při vypisování z databáze, tj. má-li útočník přístup do databáze, je jeho vložení škodlivého kódu mezi data (např. k uživatelskému jménu) neškodné.
– Další metodu bránící proti XSS je použití hlavičky *X-Content-Security-Policy: default-src 'self'* [36], ta má ale prozatím význam jen u prohlížeče Mozilla Firefox 4.0 (metoda je ve stádiu draftu, snaha o standardizaci).
– Zvýšená ochrana proti MITM (Man In The Middle) útoku zasíláním hlavičky *Strict-Transport-Security* [37], která vynutí komunikaci přes rozhraní HTTPS. Webový prohlížeč si při první návštěvě stránky uloží na jak dlouho je u dané stránky aktivní tato hlavička a při každém další návštěvě adresy oproštěné o „https://“ (tj. jen „example.com“) se nedotazuje na HTTP variantu, ale rovnou otevře HTTPS.
– Zabezpečení cookies – HttpOnly, přenos pouze přes šifrované spojení.
– Obrana před CSRF (podvržení požadavku přihlášeného uživatele) ověřováním vygenerovaného unikátního kódu u každého formuláře s každým znovunačtením stránky.
– Uživatelé s moderními prohlížeči jsou ochráněni také před útokem ClickJacking, za pomoci hlavičky `X-Frame-Options: deny`.
– Session se na serveru ukládají do SQL databáze. Není tedy problém jako v případě sdílených webových hostingů, kde se ve výchozím nastavení ukládají Session do složky `Temp` vlastněnou operačním systémem, ke které mají přístup všichni uživatelé webového serveru.
– Výpis chybových hlášení je vypnut (ukazují útočníkovi slabá místa aplikace),

ale zároveň se chyby zapisují do logovacího souboru v zabezpečené složce.
– Testovací doména je zabezpečena prostřednictvím DNSSEC, jelikož se nachází v podepsáné kořenová zóně (*.cz*).

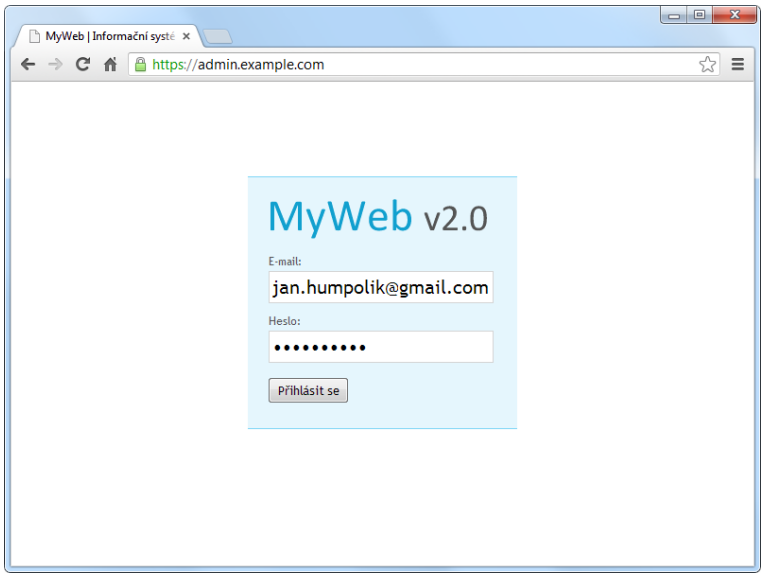



Obr. 3.4: Návštěvníci se dají blokovat podle libovolné IPv4/IPv6 „masky“.

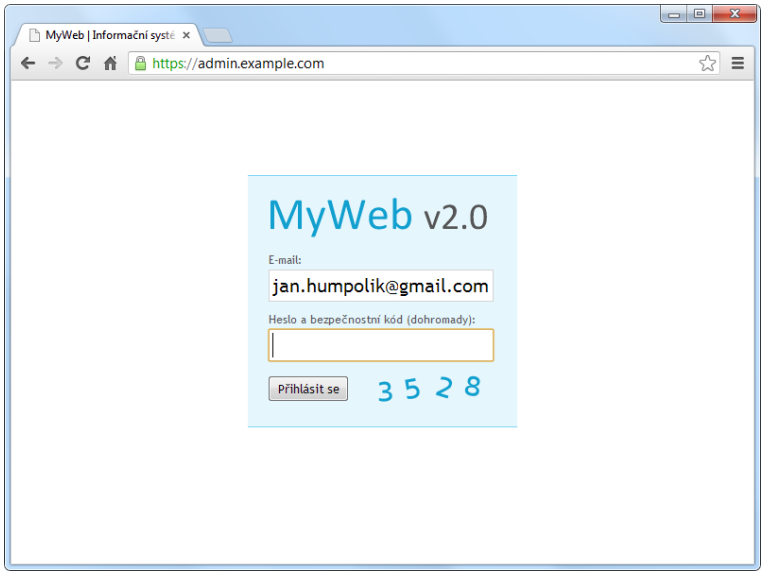
## 3.3 Vlastnosti systému

– Architektura databáze je založena na vlastním modulárním systému, který přispívá vyšší bezpečnosti a snadnější rozšiřitelnosti.
– Referenční integrita databáze zaručena pomocí *Cizích klíčů* (Foreign Keys).
– Pro hezká URL je použit `mod_rewrite`, zhoršila se tím ale přenositelnost na jiný webový server, než je dominantní Apache.
– Výstupem webové aplikace je dokument typu `text/html` (pro větší kompatibilitu) v kódování UTF-8. Využívá prvky standardů XHTML 1.0 Strict, CSS 2.1 a JavaScriptu (pro lepší *použitelnost*), je *přístupný* i v textové formě (bez CSS).
– Zabráněno opakovanému odeslání formulářových dat (POST) při obnovení webové stránky, pomocí HTTP 301/303 s přesměrováním na stejnou stránku.
– Administrátor má možnost uživatelům přenastavit uživatelům mnohé údaje, vč. nastavení vícefaktorové autentizace (obr. refadministrace). Samozřejmostí je možnost vytváření a odstraňování uživatelů.
– Jednotlivé metody vícefaktorové autentizace lze v databázi povypínat, uživatelům ale nastavení zůstane, tyto uživatele je třeba přenastavit zvlášť – nejlépe

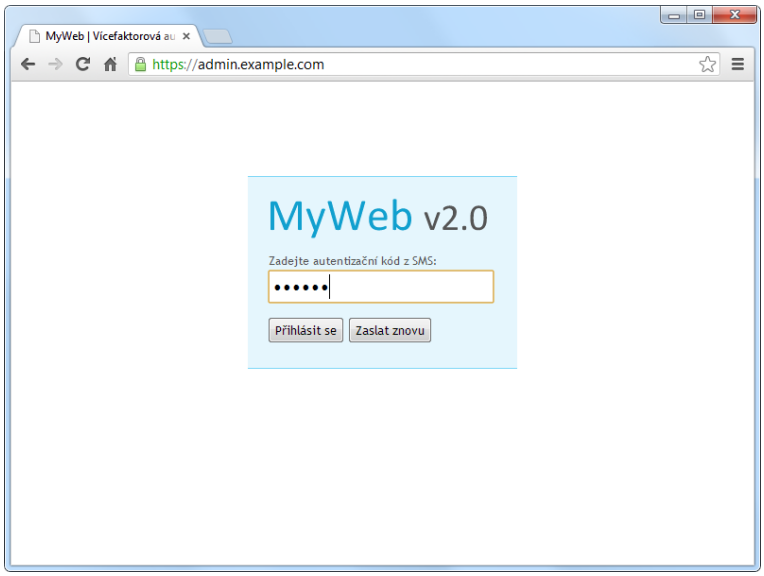
vypnout vícefaktorovou autentizaci (stejně se bez opětovného nastavení do určitých modulů nedostanou). Jde o vlastnost, lze takto nastavit např. SMS autentizaci jen uživatelům, o kterých víme, že ji nebudou zneužívat k nadměrnému zasílání a ostatním (ve skutečnosti všem) možnost SMS autentizace vypneme. Takový uživatel má možnost si zvolit i jinou metodu autentizace, ale do zapnutí vypnuté autentizace se mu již ji nepodaří nastavit zpět.



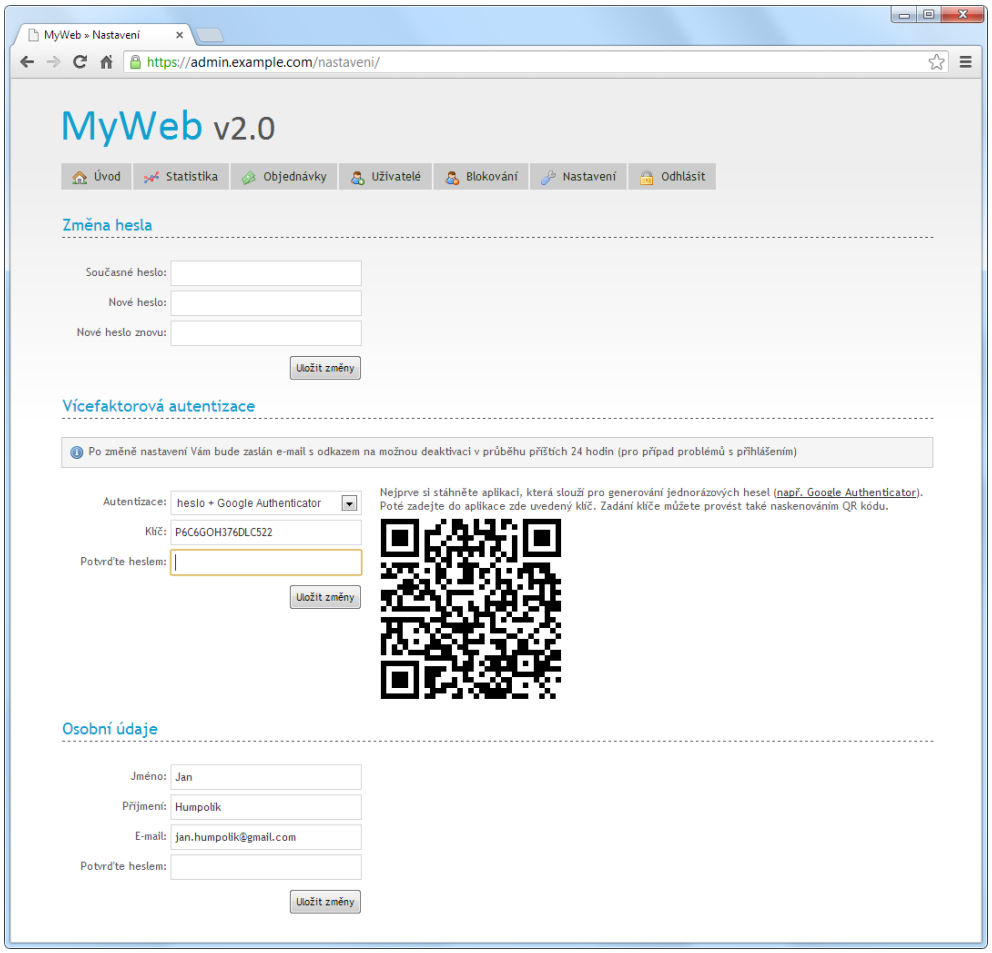

Obr. 3.5: 1. autentizační faktor (*něco vědět*) pomocí e-mailová adresy a hesla.




Obr. 3.6: V případě 3 neúspěšných pokusů se spustí systém CAPTCHA.

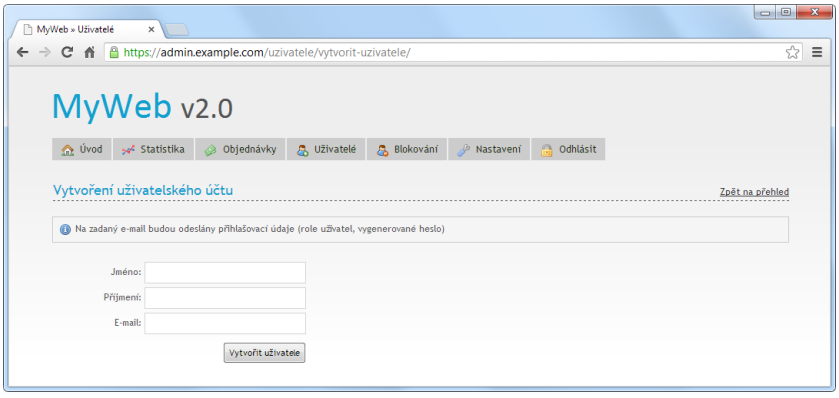



Obr. 3.7: 2. autentizační faktor (*něco mít*), v tomto případě SMS autentizace.




Obr. 3.8: Uživatelské nastavení.


Obr. 3.9: Administrátorské nastavení jiného uživatele.



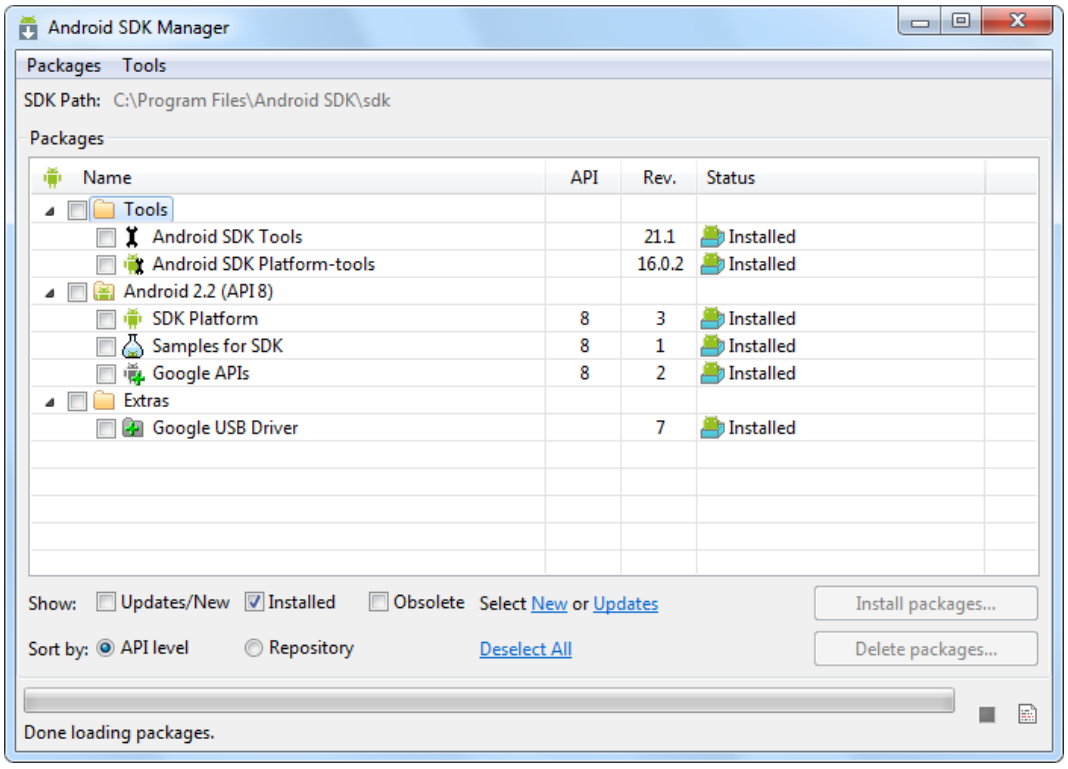

Obr. 3.10: Vytvoření uživatele, přihlašovací údaje obdrží e-mailem.

# 4 VÝVOJOVÉ PROSTŘEDÍ PRO ANDROID

Následuje návod na instalaci vývojového prostředí pro vývoj aplikací na platformě Google Android, díky čemuž je možné upravit aplikaci vyvinout v této práci.

## Postup instalace do stávajícího Eclipse

1. Stáhneme a nainstalujeme Java JDK 7 x64 (jdk-7u21-windows-x64.exe) [38].
2. Eclipse Classic 4.2.2 x64 (eclipse-SDK-4.2.2-win32-x86_64.zip) [39] rozbalíme do „C:\Program Files\Eclipse“.
3. Stáhneme Android SDK (android-sdk_r21.1-windows.zip) [40] a nainstalujeme jej do „C:\Program Files\Eclipse\android-sdk“.
4. Spustíme program Eclipse (C:\Program Files\Eclipse\eclipse.exe) a vybereme *Help > Install New Software*.
5. Do políčka *Work with* vyplníme „https://dl-ssl.google.com/android/eclipse/“.
6. Vybereme *Developer Tools* a 2x klepneme na *Next*, následně potvrdíme licenční podmínky a klepneme na *Finish*.
7. Restartujte program Eclipse.
8. Z hlavní nabídky programu Eclipse vybereme *Windows > Android SDK Manager* a nainstalujeme balíčky dle obr. 4.1.




Obr. 4.1: Výběr balíčků u instalace Android SDK.

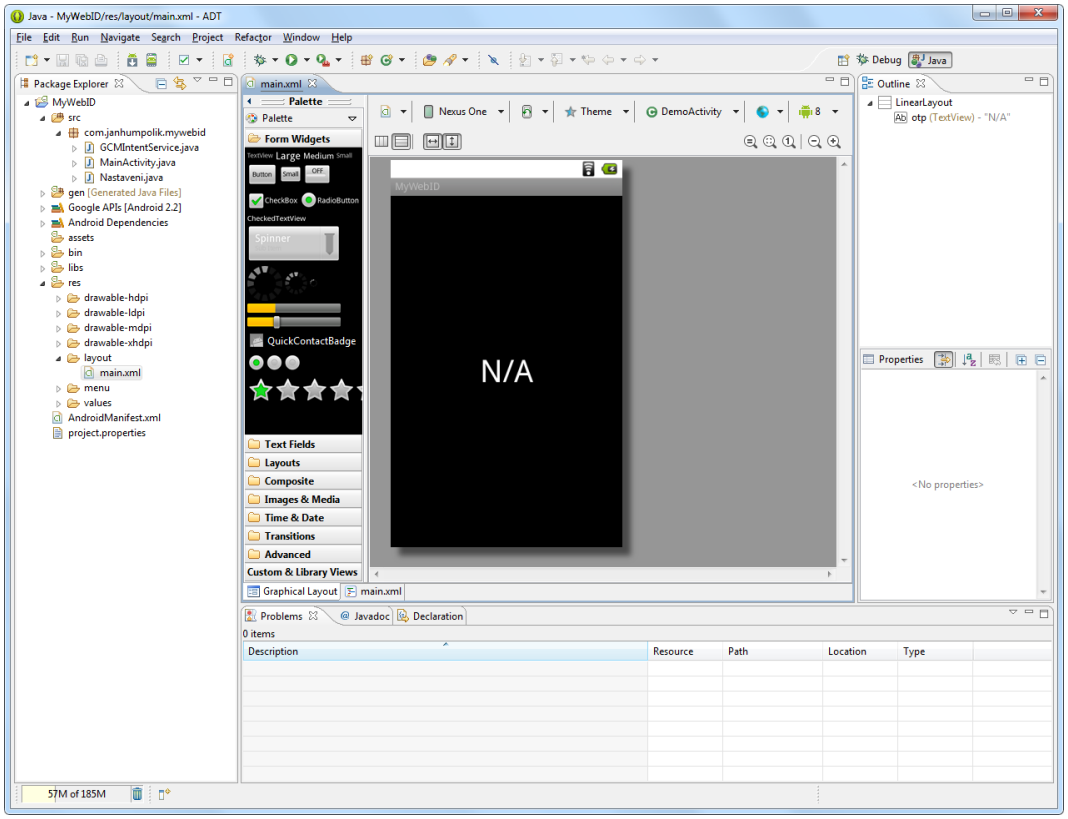
Pozn. Existuje i předpřipravený Eclipse (adt-bundle-windows-x86_64-20130219.zip) [40] přímo s rozšířením Android SDK.

## Vývoj na virtuálním zařízení

1. Z hlavní nabídky programu Eclipse vybereme *Windows* > *AVD Manager* a klepneme na tlačítko *New*.
2. Stačí vyplnit jen políčko *Name* (např. „virtual“) a pod *Target* zvolit API Level (v našem případě „Android 2.2 – API Level 8“).
3. Po vytvoření virtuálního zařízení a nějakého testovací projektu můžeme spustit virtuální zařízení tím, že klepneme na tradiční tlačítko *Run* a cca po minutě se virtuální zařízení spustí v novém okně. Mezi dalšími kompilacemi okno s virtuálním zařízením nezavíráme, spouštění tak bude během okamžiku.

## Vývoj na mobilním telefonu HTC

1. Připojte telefon pomocí USB k počítači, jako typ připojení vyberte HTC Sync.
2. Nainstalujte program HTC Sync, ten je nutný a zároveň nainstaluje i ovladače.
3. Povolte položku *Ladění USB* v *Nastavení* > *Aplikace* > *Vývoj*.




Obr. 4.2: Ukázka návrhu GUI v oficiálním vývojovém prostředí pro Android.

# 5 INSTALACE WEBOVÉHO SERVERU

Pro vyzkoušení aplikace v praktické části je potřeba mít patřičný webový server, pro návod na jeho instalaci byla zvolena nejčastější konfiguraci pro vývoj – Open Source Software na platformě Microsoft Windows.

Dále byly zvoleny instalační soubory pro HTTP server Apache z komunitního serveru Apache Lounge, tedy nikoliv přímo oficiálního webu, z důvodu integrace balíčků podpory IPv6, VC9 verze PHP (použit nový kompilátor pro vylepšení výkonu a stability) aj., které oficiální verze pro tuto platformu nemá.

Zvolený operační systém Windows Server 2012 Standard Edition x64, lze získat pod odkazem [41] zdarma [42] pro instalace na osobní počítače studentů, vyučujících či zaměstnanců školy pro potřeby výuky, není však určen na běžný školní podpůrný provoz, který s výukou přímo nesouvisí.

V bezpečnosti je třeba se držet nejnovějších verzí programů, proto i tento návod popisuje instalaci posledních verzí, pokud možno již v 64bitové variantě.

Windows Server 2012 Standard Edition x64 [41]
Apache 2.4.4 [43] (httpd-2.4.4-win32.zip)
OpenSSL 1.0.1e – součástí instalace Apache HTTP Server
MariaDB 5.5.31 x64 [44] (mariadb-5.5.31-winx64.zip)
PHP 5.4.15 [45] (php-5.4.15-Win32-VC9-x86.zip)
phpMyAdmin 4.0.2 [46] (phpMyAdmin-4.0.2-all-languages.zip)

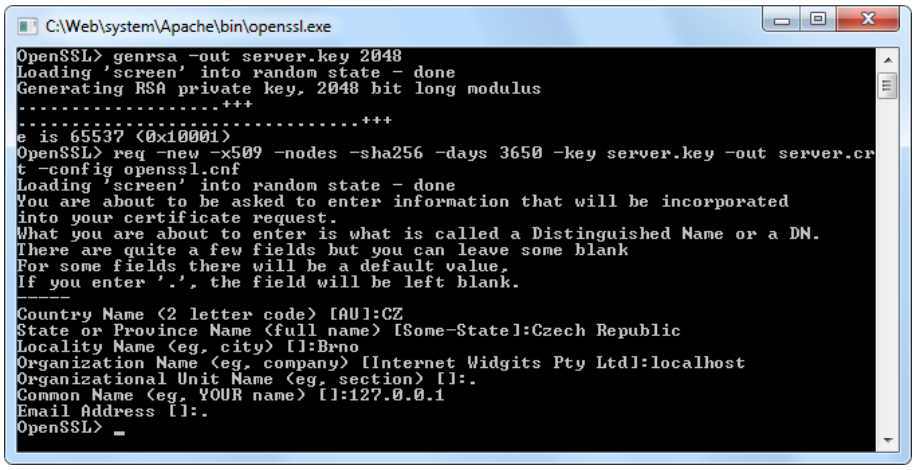
## Postup instalace

Byla zvolena instalace s následující strukturou, v případě použití jiné, je třeba konfiguraci podle toho upravit. Jde o umístění přímo v kořenovém adresáři, protože server Apache vyžaduje oprávnění „Zobrazení obsahu složky“ už odtud. Lze tedy využít i jiné, více vnořené, složky (např. Program Files), ale Apache (a tím i PHP skripty – v případě modulu) budou moci zobrazit seznam souborů v těchto složkách (např. ve všech dalších složkách v Program Files). Názvy vnořených složek nesmí obsahovat háčky a čárky, mezera je možná.

C:\Web\htdocs
C:\Web\system\Apache
C:\Web\system\MySQL
C:\Web\system\PHP
C:\Web\system\phpMyAdmin

1. Vzhledem k tomu, že server Apache nemůže sdílet stejný port s jinou TCP/IP aplikací, je třeba zastavit či přenastavit aplikace využívající porty 80 a 443.
2. Stáhneme instalační soubory z internetu pod odkazy v úvodní části tohoto návodu. Instalační balík pro první instalaci je k nalezení pod odkazem [43], atd. postupujeme.
3. Rozbalíme instalační balík souborů serveru Apache (httpd-2.4.4-win32.zip) a obsah složky Apache2 přesuneme do C:\Web\system\Apache.
4. V textovém editoru otevřeme soubor „Apache\conf\httpd.conf“ a provedeme následující nahrazení. Případně využijeme předpřipravených konfiguračních souborů v příloze. Čísla označují čísla řádků v konfiguračním souboru, jejich umístění se může s novými verzemi lišit. Tip: Pro zobrazení čísla řádku v aplikaci Poznámkový blok je nutné zrušit Zalamování řádků a povolit Stavový řádek.
   `026: ServerSignature Off` *(omezená identifikace serveru)*
   `027: ServerTokens Prod` *(omezená identifikace serveru)*
   `037: ServerRoot "C:/Web/system/Apache"`
   `038: PHPIniDir "C:/Web/system/PHP/"`
   `059: Listen 443`
   `149: LoadModule rewrite_module modules/mod_rewrite.so`
   `162: LoadModule ssl_module modules/mod_ssl.so`
   `172: LoadModule php5_module "C:/Web/system/PHP/php5apache2_4.dll"`
   `213: ServerName SERVER`
   `242: DocumentRoot "C:/Web/htdocs"`
   `243: <Directory "C:/Web/htdocs»`
   `263: AllowOverride All` *(povolení konfiguračních souborů .htaccess)*
   `276: DirectoryIndex index.html index.htm index.php`
   `359: Alias /db "C:/Web/system/phpMyAdmin/"`
   `375: <Directory "C:/Web/system/phpMyAdmin/»`
   `376: AllowOverride All`
   `405: AddType application/x-httpd-php .php`
   `406: AddType application/x-httpd-php-source .phps`
   `509: Include conf/extra/httpd-ssl.conf`
5. Otevřeme soubor „Apache\conf\extra\httpd-ssl.conf“ a v něm změníme:
   `086: DocumentRoot "C:/Web/htdocs"`
6. Vytvoříme složku „C:\Web\htdocs“.
7. V souboru „Apache\conf\openssl.cnf“ změníme následující řádek:
   `240: basicConstraints = CA:false` *(certifikační úřad nebo koncová entita)*
   A následně soubor zkopírujeme do „Apache\bin“.
8. Spustíme „Apache\bin\openssl.exe“ (obr. 5.1) a zadáme následující 2 příkazy:
   `genrsa -out server.key 2048`

```
req -new -x509 -nodes -sha256 -days 3650 -key server.key \
    -out server.crt -config openssl.cnf
```




Obr. 5.1: Nastavení OpenSSL.

9. Soubory server.key a server.crt z „Apache\bin“ přesuneme do „Apache\conf“.
10. PHP ve verzi VC9 Thread Safe (php-5.4.15-Win32-VC9-x86.zip) rozbalíme do „C:\Web\system\PHP“.
11. V této složce přejmenujeme soubor „php.ini-production“ na „php.ini“ a v něm provedeme následující úpravy:

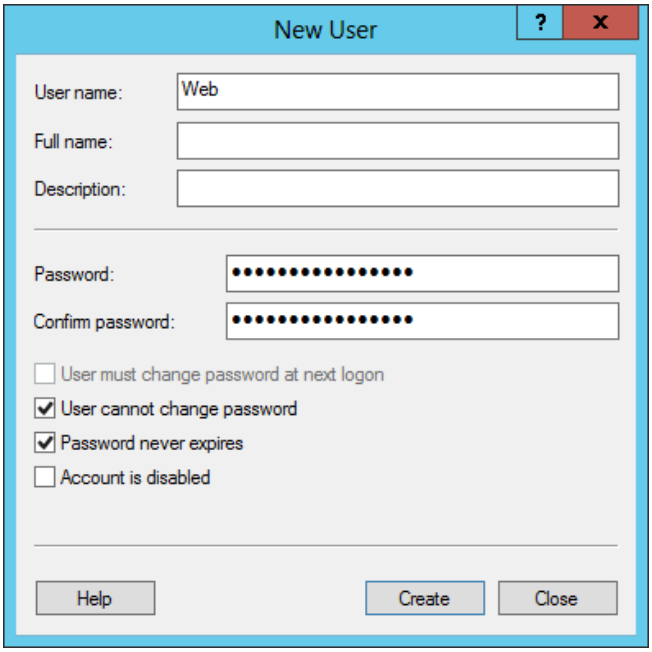
    ```
    308: open_basedir = "C:/Web/htdocs/;C:/Users/Web/AppData/Local/ \
         Temp/;C:/Web/system/phpMyAdmin/"
    375: expose_php = Off
    461: error_reporting = E_ALL & ~E_NOTICE
    723: extension_dir = "C:/Web/system/PHP/ext/"
    868: extension=php_curl.dll (knihovna pro pro přenos dat po HTTP, FTP atp.)
    870: extension=php_gd2.dll (knihovna pro generování obrázků)
    877: extension=php_mbstring.dll (funkce pro multibytové řetězce, např. UTF-8)
    880: extension=php_mysqli.dll (připojení k databázi MySQL)
    883: extension=php_openssl.dll (práce s certifikáty)
    918: date.timezone = Europe/Prague
    1412: session.name = ID (změna názvů relací, původně PHPSESSID)
    ```

12. Přes příkazový řádek (klávesa Win+R „cmd“) spustíme následující příkaz: „C:\Web\system\Apache\bin\httpd.exe -k install“. Tímto nainstalujeme Apache server jako službu operačního systému Windows. Pro případnou odinstalaci slouží přepínač „httpd.exe -k uninstall“.
13. MariaDB (mariadb-5.5.31-winx64.zip) rozbalíme do složky „C:\Web\system\MariaDB“ a vložíme soubor „my.cnf“ z přílohy DP.

14. Opět přes příkazový řádek spustíme instalaci služby:
„C:\Web\system\MariaDB\bin\mysqld –install MariaDB
–defaults-file="C:\Web\system\MariaDB\my.ini"“.
V případě odinstalace je možné využít přepínač „–remove MariaDB“.
15. phpMyAdmin (phpMyAdmin-4.0.2-all-languages.zip) rozbalíme do složky „C:\Web\system\phpMyAdmin“.
16. Tam také vytvoříme soubor „config.inc.php“, do něhož vložíme nastavení:

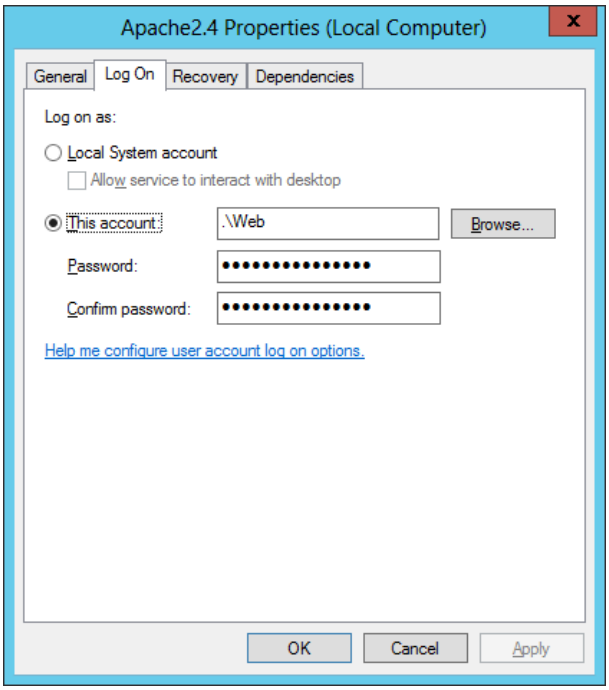
```
<?php
$cfg['Servers'][1]['verbose'] = 'SERVER';
$cfg['Servers'][1]['host'] = '.';
$cfg['Servers'][1]['auth_type'] = 'cookie';
$cfg['Servers'][1]['extension'] = 'mysqli';
$cfg['Servers'][1]['hide_db'] = '(mysql|information_schema \
     |performance_schema)';
$cfg['Servers'][1]['AllowNoPassword'] = TRUE;
$cfg['Servers'][1]['ssl'] = FALSE;
$cfg['Export']['compression'] = 'zip';
$cfg['Export']['method'] = 'custom';
$cfg['Export']['sql_disable_fk'] = TRUE;
$cfg['Export']['sql_drop_database'] = TRUE;
$cfg['Export']['sql_drop_table'] = FALSE;
$cfg['Export']['sql_include_comments'] = FALSE;
$cfg['Export']['sql_procedure_function'] = TRUE;
$cfg['Export']['sql_use_transaction'] = TRUE;
$cfg['AllowUserDropDatabase'] = TRUE;
$cfg['blowfish_secret'] = 'W6NyCR2AO9tCC2Sq';
$cfg['collation_connection'] = 'utf8_czech_ci';
$cfg['DefaultLang'] = 'cs';
$cfg['EditInWindow'] = FALSE;
$cfg['ForceSSL'] = TRUE;
$cfg['lang'] = 'cs';
$cfg['PmaNoRelation_DisableWarning'] = TRUE;
$cfg['RememberSorting'] = FALSE;
$cfg['RepeatCells'] = 0;
$cfg['RetainQueryBox'] = TRUE;
$cfg['ServerDefault'] = 1;
$cfg['ShowPhpInfo'] = TRUE;
$cfg['VersionCheck'] = FALSE;
?>
```

17. Proměnnou `$cfg['blowfish_secret']` naplníme nějakým jiným náhodným řetězcem, je to nastavení šifrovacího klíče pro cookies.
18. Můžete omezit přístup k administraci databáze na IP adresy – vytvořením souboru „.htaccess“ ve složce „system\phpMyAdmin“ s následujícím obsahem:
    `deny from all`
    `allow from fc00::/64` *(je třeba doplnit používaný IPv6 prefix síťě)*
    `allow from 10.0.0.` *(je třeba doplnit používaný IPv4 prefix síťě)*
19. Nyní vytvoříme oprávnění na úrovni operačního systému (s menšími pravomocemi), aby webový server neměl tak velké oprávnění, jako uživatel pod kterým byl nainstalován (typicky administrátorská). Spustíme program `lusrmgr.msc` (např. přes klávesovou zkratku `Win+R`) a ve složce *Users* vytvoříme nového uživatele „Web“ (obr. 5.2). V podrobnostech nezapomeneme nastavit volby *User must change password at next logon*, *User cannot change password* a *Password never expires*.

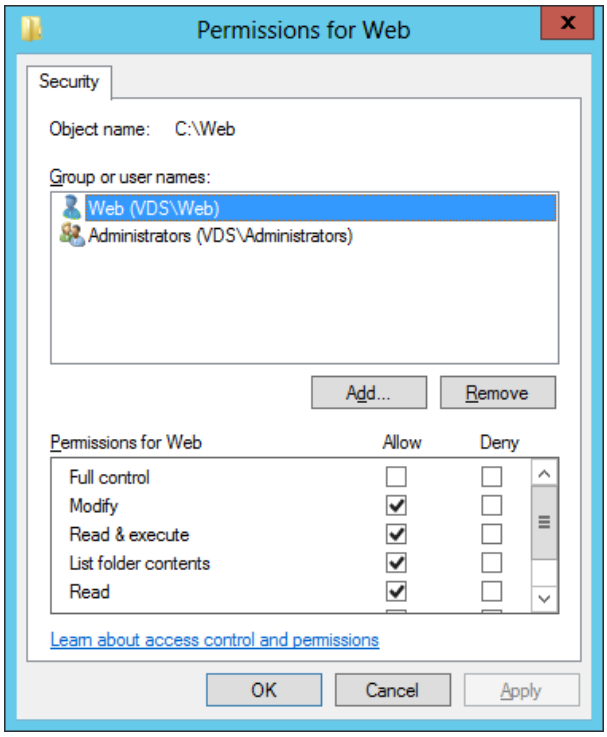



Obr. 5.2: Vytvoření uživatele Web.

20. Následně nově vytvořeného uživatele připojíme přes *Služby* (`services.msc`) službám „Apache2.4“ a „MariaDB“ (obr. 5.3).
21. Podle obr. 5.4 ponecháme jen uživatele *Administrator* a přidáme uživatele *Web*, kterému dáme téměř plná oprávnění pro danou složku.
22. Tímto je instalace hotova, nejsnadnější možnost otestování všech součástí najednou je přihlášení se do správy databází přes HTTPS připojení. Adresa je v tomto případě `https://127.0.0.1/db/` (doporučuji použít „127.0.0.1“, než „localhost“), uživatelské jméno pro přihlášení je „root“ (bez hesla).



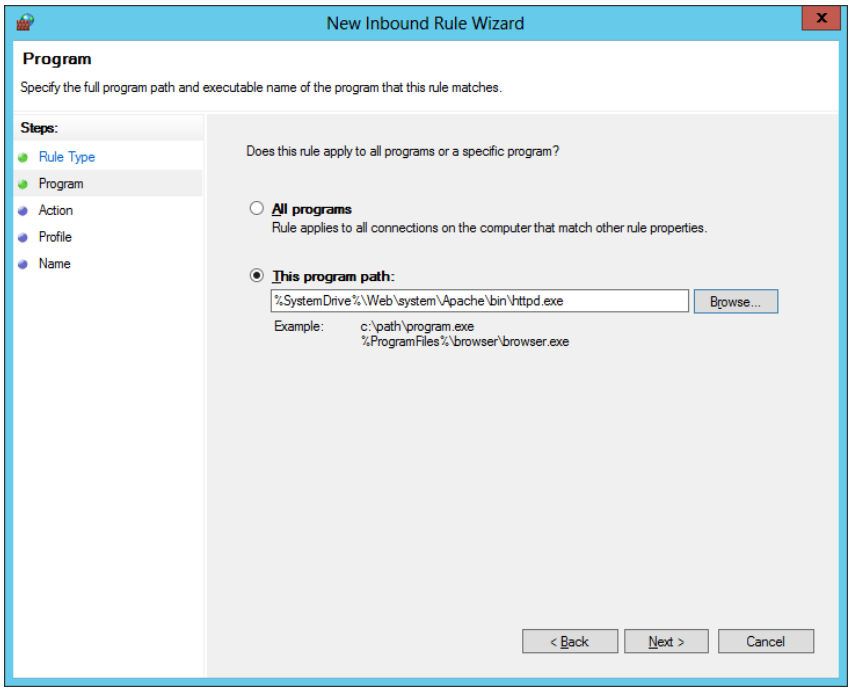

Obr. 5.3: Změna spouštění služby Apache, podobně tak služba MariaDB.




Obr. 5.4: Vytvoření oprávnění pro složku Web.

23. Pro využití webového serveru v internetu je většinou nutné nastavit směrování portů TCP 80 a 443 na NATu (Network Address Translation), a také povolit tyto porty na

firewallu. V případě použití vestavěného firewallu ve Windows Serveru je nejrychlejší vyhledat přes klávesovou zkratku `Win+Q` program *Windows Firewall with Advanced Security* a v něm povolit „C:\Web\system\Apache\bin\httpd.exe“ (obr. 5.5).




Obr. 5.5: Povolení aplikace Apache ve Windows Firewallu.

# 6 BEZPEČNOSTNÍ AUDIT SYSTÉMU

V této kapitole jsou rozebrány slabiny metod, které byly v průběhu psaní nalezeny.

## 6.1 MyWebID

Náchylnost na *malware*, který by mohl přeposílat autentizační kódy útočníkovi.
A případná kompromitace GCM serverů, přes které se autentizační kódy zasílají, ty jsou ve vlastnictví a správě společnosti Google.

## 6.2 Google Authenticator

V případě vyzrazení/získání klíče, podle kterého TOTP algoritmus vypočítává autentizační kódy, se uživatel nedozví, že byl kompromitován. Také náchylnost na *malware*.

## 6.3 SMS autentizace

Když pomineme, že zprávy může útočník zachytit dešifrováním datových přenosů v mobilních sítích GSM/UMTS, nebo že může mít přístup do SMS brány mobilního operátora, tak existuje ještě jedna, mnohem snazší, možnost.

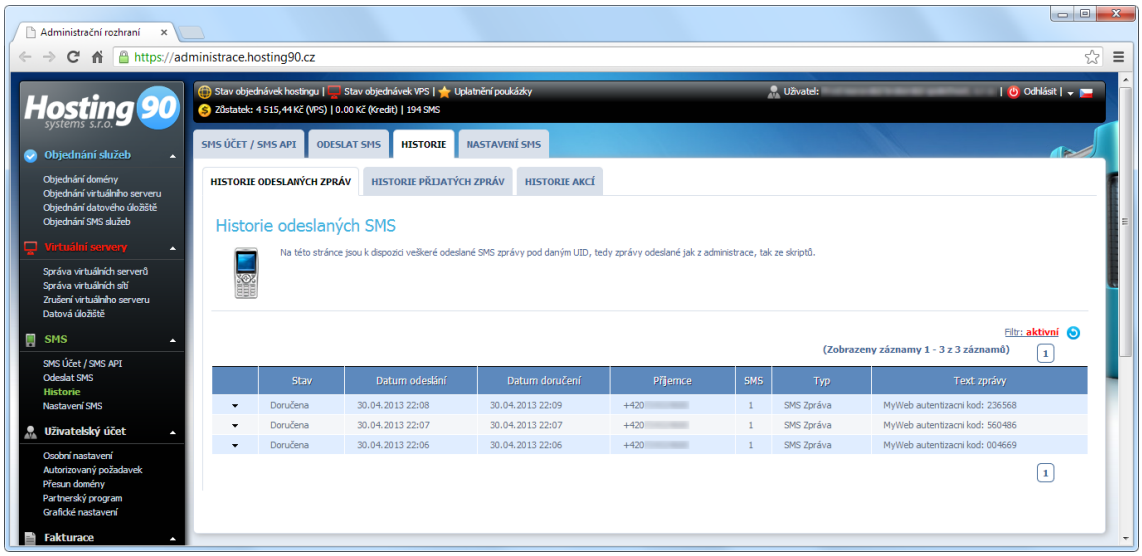
API od společnosti Hosting90, které je využito v této implementaci, obsahuje administrační stránku, která obsahuje i historii všech odeslaných SMS (obr. 6.1). Do této administrace [49] se přihlašuje jen pomocí uživatelského jména a hesla (přes šifrované spojení). Jestliže by útočník získal (např. HW keyloggerem) přístup do této administrace, dozvěděl by se SMS autentizační kódy v časovém limitu.

Pro chytré mobilní telefony je i zde problém náchylnosti na *malware*.

## 6.4 YubiKey

V praktické části je použit autentizační server YubiCloud, místo vytvoření vlastního autentizačního serveru (z důvodu, že není nutné upravovat/flashovat konfiguraci každého jednotlivého YubiKey a že je takto možné konkrétní YubiKey využit ve více projektech zároveň). YubiCloud ale, jak název napovídá, není na našem serveru, ale někde v *cloudu*, pod jinou subdoménou, resp. přímo pod jinou doménou.

Všechny autentizační požadavky na YubiCloud jsou standardně typu HTTP GET (příslušně v URL kódovány v base64). Viz „Validation Protocol Version 2.0“ [50]. Každý zaslaný požadavek je podepsán. K ověření odpovědi od YubiCloud (jestli




Obr. 6.1: Historie odeslaných SMS.

nebyla pozměněna po cestě) klient (náš webový server) ověřuje HMAC podpis, nebo použije HTTPS spojení (a ověří certifikát).

Tím je docíleno, že v průběhu psaní této práce, nebyly nalezeny slabá místa zabezpečení. Alespoň v případě této PHP implementace, ta je navíc nastavena na komunikaci rovnou přes HTTPS GET.

## 6.5 Certifikát (čipová karta)

Je třeba uchránit soukromý klíč před prozrazením, nejlépe umístěním certifikátu (společně se soukromým klíčem) do čipové karty/USB tokenu.

# 7 ZÁVĚR

V této diplomové práci byl proveden teoretický úvod, návrh a vývoj vícefaktorové autentizace (aplikace MyWebID a implementace 4 dalších metod), nakonec byly metody začleněny do navržené webové aplikace MyWeb, která obsahuje další nápady na zlepšení zabezpečení.

Co se týče vlastní aplikace pro operační systém Android (MyWebID), tak v případě, že budeme počítat s nástupem služebních telefonů s tímto systémem (obměnou stávajících telefonů ve firmě), nemusíme tuto formu autentizace brát jako dodatečný náklad, jako tomu je v případě alternativních technik (např. RSA SecurID). Navíc nemusí jít přímo o mobilní telefon, ale i tablet s tímto operačním systémem. Připojení k internetu nemusí být nutně přes mobilního operátora za měsíční poplatek, ale přes vnitrofiremní Wi-Fi síť, která by měla pro tyto zařízení vlastní SSID (kvůli časté nemožnosti použít na těchto zařízeních IEEE 802.1X) s aktivní funkcí Clients Isolation a omezením jen na Google Play. Aplikace MyWebID je z důvodů využití GCM (Google Cloud Messaging) a jejího vývoje (jako nativní aplikaci v Java) závislá na platformě Android, v budoucnu však bude možné aplikaci portovat i pro jiné platformy. Nemusí jít přímo znovu o vývoj nativních aplikací (iOS, Windows Phone, Blackberry), ale v případě podpory GCM i pro jiné platformy (aktuálně Android a iOS), by v budoucnu mohlo dojít k urychlení vývoje použitím frameworků (např. PhoneGap, Apache Cordova, Sencha Touch, Appcelerator Titanium).

Jako optimální metoda 2. autentizačního faktoru se dle mého názoru jeví 2 možné varianty: certifikát vydaný CA jen pro tuto činnost, jenž je uložen vč. soukromého klíče ve webovém prohlížeči na uživatelově osobním počítači (který má šifrované pevné disky a používá vyšší formu zabezpečení pro přihlášení do operačního systému) – pro ergonomii a rychlost použití, nebo YubiKey – pro jeho obranyschopnost proti škodlivému software (malware) a jeho relativně nízkou cenu ve srovnání s čtečkami čipových karet a USB tokeny.

# LITERATURA


[1] *Wired : Google Declares War on the Password* [online]. 2013 [cit. 2013-05-27]. Dostupný z WWW: <http://www.wired.com/wiredenterprise/2013/01/google-password/all/>.

[2] *Lupa : Dvoufaktorovou autentizaci Twitteru může útočník vypnout* [online]. 2013 [cit. 2013-05-27]. Dostupný z WWW: <http://www.lupa.cz/clanky/dvoufaktorovou-autentizaci-twitteru-muze-utocnik-vypnout/>.

[3] *Dashboards | Android Developers* [online]. 2013 [cit. 2013-04-26]. Dostupný z WWW: <http://developer.android.com/about/dashboards/index.html>.

[4] *Engadget : Google now at 1.5 million Android activations per day* [online]. 2013 [cit. 2013-04-25]. Dostupný z WWW: <http://www.engadget.com/2013/04/16/eric-schmidt-google-now-at-1-5-million-android-activations-per/>.

[5] *Engadget : Google now at 1.5 million Android activations per day* [online]. 2013 [cit. 2013-04-25]. Dostupný z WWW: <http://www.engadget.com/2013/03/13/andy-rubin-leaves-google-sundar-pichai-to-lead-android/>.

[6] *Google Cloud Messaging for Android* [online]. 2013 [cit. 2013-01-25]. Dostupný z WWW: <http://developer.android.com/google/gcm/index.html>.

[7] *Web App Security : Google TOTP Two-factor Authentication for PHP* [online]. 2013 [cit. 2013-05-27]. Dostupný z WWW: <http://www.idontplaydarts.com/2011/07/google-totp-two-factor-authentication-for-php/>.

[8] *GitHub : Google Authenticator PHP class* [online]. 2013 [cit. 2013-05-27]. Dostupný z WWW: <https://github.com/PHPGangsta/GoogleAuthenticator>.

[9] *Hosting90 : Dokumentace k API (SMS funkce)* [online]. 2013 [cit. 2013-05-27]. Dostupný z WWW: <https://www.hosting90.cz/napoveda-dokumentace-k-sms-api>.

[10] *Twilio SMS Pricing* [online]. 2013 [cit. 2013-05-27]. Dostupný z WWW: <http://www.twilio.com/sms/pricing>.

[11] *Yubico : Static Password* [online]. c2013 [cit. 2013-04-21]. Dostupný z WWW: <http://www.yubico.com/products/services-software/personalization-tools/static-password/>.

[12] *Yubico : YubiKey NEO* [online]. c2013 [cit. 2013-04-21]. Dostupný z WWW: <http://www.yubico.com/products/yubikey-hardware/yubikey-neo/>

[13] *Yubico : YubiKey Nano* [online]. c2013 [cit. 2013-04-21]. Dostupný z WWW: <http://www.yubico.com/products/yubikey-hardware/yubikey-nano/>.

[14] *Yubico : Images* [online]. c2013 [cit. 2013-04-21]. Dostupný z WWW: <http://www.yubico.com/press/images/>.

[15] *Yubikey v2.0* [online]. 2009 [cit. 2013-04-21]. Dostupný z WWW: <http://www.pepak.net/bezpecnost/yubikey-v2-0/>.

[16] *Yubico : The YubiKey Manual* [online]. 2012 [cit. 2013-04-21]. Dostupný z WWW: <http://www.yubico.com/wp-content/uploads/2012/10/YubiKey-Manual-v2.2.pdf>.

[17] *Yubico : YubiKey Technical description* [online]. c2013 [cit. 2013-04-26]. Dostupný z WWW: <http://www.yubico.com/products/yubikey-hardware/yubikey/technical-description/>.

[18] *Yubico : YubiCloud Technical description* [online]. c2013 [cit. 2013-04-26]. Dostupný z WWW: <http://www.yubico.com/products/services-software/yubicloud/technical-description/>.

[19] *Yubico : Web Store* [online]. c2013 [cit. 2013-04-28]. Dostupný z WWW: <https://store.yubico.com/>.

[20] *Yubico : Configure your YubiKey* [online]. c2013 [cit. 2013-04-28]. Dostupný z WWW: <http://www.yubico.com/products/services-software/personalization-tools/use/>.

[21] *Yubico : AES Key Upload* [online]. c2013 [cit. 2013-04-28]. Dostupný z WWW: <http://www.yubico.com/products/services-software/personalization-tools/aes-key-upload/>.

[22] *Yubico : Get API Key* [online]. c2013 [cit. 2013-04-28]. Dostupný z WWW: <https://upgrade.yubico.com/getapikey/>.

[23] *GitHub – php-yubico* [online]. 2013 [cit. 2013-04-28]. Dostupný z WWW: <https://github.com/Yubico/php-yubico>.

[24] *PEAR* [online]. 2011 [cit. 2013-04-28]. Dostupný z WWW: <http://pear.php.net/package/PEAR/download>.

[25] *Přihlašování čipovými kartami – úvod do problematiky* [online]. 2013 [cit. 2013-04-27]. Dostupný z WWW: <http://www.sevecek.com/Lists/Posts/Post.aspx?ID=295>.

[26] *Aliance Fido chce vymýtit textová hesla na webu* [online]. 2013 [cit. 2013-04-27]. Dostupný z WWW: <http://www.zive.cz/bleskovky/aliance-fido-chce-vymytit-textova-hesla-na-webu/sc-4-a-168590/default.aspx>.

[27] *How FIDO Works* [online]. c2013 [cit. 2013-04-27]. Dostupný z WWW: <http://www.fidoalliance.org/how-it-works.html>.

[28] *Aliance FIDO přichází s novým autentizačním protokolem OSTP* [online]. 2013 [cit. 2013-04-27]. Dostupný z WWW: <http://businessworld.cz/novinky/aliance-fido-prichazi-s-novym-autentizacnim-protokolem-10460>.

[29] *mojeID* [online]. 2013 [cit. 2013-04-27]. Dostupný z WWW: <http://www.mojeid.cz/>.

[30] *StartSSL Certificates & Public Key Infrastructure* [online]. c2004-2013 [cit. 2013-04-27]. Dostupný z WWW: <http://www.startssl.com/?app=0>.

[31] *StartSSL Comparison Chart* [online]. c2004-2013 [cit. 2013-04-27]. Dostupný z WWW: <https://www.startssl.com/?app=40>.

[32] *SSL v ohrožení: komunikaci je možné dešifrovat* [online]. 2011 [cit. 2013-04-27]. Dostupný z WWW: <http://www.root.cz/clanky/ssl-v-ohrozeni-komunikaci-je-mozne-desifrovat/>.

[33] *YouTube : BEAST vs HTTPS* [online]. 2011 [cit. 2013-04-27]. Dostupný z WWW: <http://www.youtube.com/watch?v=BTqAIDVUvrU>.

[34] *Ivan Ristić: Mitigating the BEAST attack on TLS* [online]. 2011 [cit. 2013-04-27]. Dostupný z WWW: <http://blog.ivanristic.com/2011/10/mitigating-the-beast-attack-on-tls.html>.

[35] *Qualys SSL Labs : SSL Server Test* [online]. c2009-2013 [cit. 2013-04-27]. Dostupný z WWW: <https://www.ssllabs.com/ssldb/analyze.html>.

[36] *Content Security Policy* [online]. c2011 [cit. 2013-04-27]. Dostupný z WWW: <http://www.soom.cz/data/ContentSecurityPolicy.pdf>.

[37] *Jak vynutit přenos po HTTPS pomocí hlavičky* [online]. 2010 [cit. 2013-04-27]. Dostupný z WWW: <http://www.zdrojak.cz/zpravicky/jak-vynutit-prenos-po-https-pomoci-hlavicky/>.

[38] *Java SE Downloads* [online]. 2013 [cit. 2013-05-01]. Dostupný z WWW: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

[39] *Eclipse Downloads* [online]. c2013 [cit. 2013-05-01]. Dostupný z WWW: <http://www.eclipse.org/downloads/>.

[40] *Android SDK | Android Developers* [online]. 2013 [cit. 2013-05-01]. Dostupný z WWW: <http://developer.android.com/sdk/index.html>.

[41] *Microsoft DreamSpark : Product Windows Server 2012* [online]. c2013 [cit. 2013-01-25]. Dostupný z WWW: <https://www.dreamspark.com/Product/Product.aspx?productid=42>.

[42] *MSDN : Microsoft Dreamspark — pro všechny studenty a učitele v ČR* [online]. c2011 [cit. 2013-04-27]. Dostupný z WWW: <http://blogs.msdn.com/b/vyvojari/archive/2011/05/17/microsoft-dreamspark-pro-vsechny-studenty-a-ucitele-v-cr.aspx>.

[43] *Apache on Windows 32 binaries and modules download* [online]. [2013] [cit. 2013-01-25]. Dostupný z WWW: <http://www.apachelounge.com/download/>.

[44] *Downloads - MariaDB* [online]. c2013 [cit. 2013-01-25]. Dostupný z WWW: <https://downloads.mariadb.org/>.

[45] *PHP For Windows : Binaries and sources Releases* [online]. c2001-2013 [cit. 2013-01-25]. Dostupný z WWW: <http://windows.php.net/download/>.

[46] *phpMyAdmin : Download* [online]. c2003-2013 [cit. 2013-01-25]. Dostupný z WWW: <http://www.phpmyadmin.net/home_page/downloads.php>.

[47] *PKI ve Windows* [online]. 2011 [cit. 2013-04-13]. Dostupný z WWW: <http://www.sevecek.com/Lists/Posts/Post.aspx?ID=31>.

[48] *NIST SP 800-57, Recommendation for Key Management* [online]. 2007 [cit. 2013-04-13]. Dostupný z WWW: <http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf>.

[49] *Hosting90 : Administrační rozhraní* [online]. 2013 [cit. 2013-05-26]. Dostupný z WWW: <https://administrace.hosting90.cz/>.

[50] *Yubico : ValidationProtocolV20* [online]. 2012 [cit. 2013-04-24]. Dostupný z WWW: <https://github.com/Yubico/yubikey-val/wiki/ValidationProtocolV20>.

# SEZNAM ZKRATEK

| | |
|---|---|
| AES | Advanced Encryption Standard |
| API | Application Programming Interface |
| BEAST | Browser Exploit Against SSL/TLS |
| CA | Certification Authority |
| CAPTCHA | Completly Automated Public Turing test to tell Computers and Humans Apart |
| DES | Data Encryption Standard |
| D-H | Diffie–Hellman |
| DNSSEC | Domain Name System Security Extensions |
| DP | Diplomová práce |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| EV | Extended Validation |
| FIDO | Fast IDentity Online |
| GCM | Google Cloud Messaging |
| GUI | Graphical User Interface |
| HOTP | HMAC-based One Time Password |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| JRE | Java Runtime Environment |
| MITM | Man In The Middle |
| NAT | Network Address Translation |
| NFC | Near Field Communication |
| NIST | National Institute of Standards and Technology |
| NSA | National Security Agency |

| | |
|---|---|
| OCSP | Online Certificate Status Protocol |
| OS | Operační systém |
| OSTP | Online Security Transaction Protocol |
| PAM | Pluggable Authentication Module |
| PHP | PHP: Hypertext Preprocessor |
| PKI | Public Key Infrastructure |
| RFC | Request for Comments |
| RSA | Rivest, Shamir, Adleman |
| SHA | Secure Hash Algorithm |
| SMS | Short Message Service |
| SQL | Structured Query Language |
| sshd | OpenSSH Daemon |
| SSL | Secure Sockets Layer |
| SSO | Single Sign-On |
| TLS | Transport Layer Security |
| TOTP | Time-based One-time Password Algorithm |
| TPM | Trusted Platform Module |
| UI | User Interface |
| URL | Uniform Resource Locator |
| USB | Universal Serial Bus |
| UX | User Experience |
| WWW | World Wide Web |
| 2TDEA | Double DES |
| 3TDEA | Triple DES |

# SEZNAM PŘÍLOH

# A OBSAH PŘILOŽENÉHO CD

**• Zdrojové texty webové aplikace MyWeb**

Návod na jejich zprovoznění:

0. Pokud máte tu možnost, využijte návodu k instalaci a nastavení webového serveru z kap. 5.
1. Importujte (např. pomocí nástroje phpMyAdmin) do předem vytvořené databáze (řazení `utf8_czech_ci`) obsah souboru „databaze.sql“.
2. V souboru „index.php“ nastavte připojení k databázovému serveru.
3. Ve stejném souboru je také třeba nastavit GCM, SMS a Yubico API.
4. Nyní se můžete přihlásit, e-mail: `root@example.com`, heslo: `root`.

**• Zdrojové texty a knihovny autentizačních metod**

- MyWebID (pro zprovoznění aplikace můžete využít návodu Vývojové prostředí pro Android, z kap. 4).
- Google Authenticator „proof of concept“.
- SMS autentizace „proof of concept“.
- YubiKey „proof of concept“.
- Certifikát „proof of concept“.

**• Konfigurační soubory webového serveru**

Pro návod z kapitoly 5 – Instalace webového serveru.

**• Elektronická verze této práce ve formátu PDF**

# B VELIKOST RAINBOW TABLES

| Délka | Kombinací [mil.] | Velikost Rainbow tables [GiB] | | |
|---|---|---|---|---|
| | | SHA1 | SHA256 | SHA512 |
| 5 | 12 | 0 | 0 | 1 |
| 6 | 309 | 6 | 9 | 18 |
| 7 | 8 032 | 150 | 239 | 479 |
| 8 | 208 827 | 3 890 | 6 224 | 12 447 |
| 9 | 5 429 504 | 101 132 | 161 812 | 323 624 |
| 10 | 141 167 096 | 2 629 442 | 4 207 107 | 8 414 215 |

Tab. B.1: Rainbow tables: Znaky z množin a–z (celkem 26 znaků).

| Délka | Kombinací [mil.] | Velikost Rainbow tables [GiB] | | |
|---|---|---|---|---|
| | | SHA1 | SHA256 | SHA512 |
| 5 | 60 | 1 | 2 | 4 |
| 6 | 2 177 | 41 | 65 | 130 |
| 7 | 78 364 | 1 460 | 2 335 | 4 671 |
| 8 | 2 821 110 | 52 547 | 84 076 | 168 151 |
| 9 | 101 559 957 | 1 891 702 | 3 026 723 | 6 053 445 |
| 10 | 3 656 158 440 | 68 101 258 | 108 962 013 | 217 924 025 |

Tab. B.2: Rainbow tables: Znaky z množin a–z, 0–9 (celkem 36 znaků).

| Délka | Kombinací [mil.] | Velikost Rainbow tables [GiB] | | |
|---|---|---|---|---|
| | | SHA1 | SHA256 | SHA512 |
| 5 | 916 | 17 | 27 | 55 |
| 6 | 56 800 | 1 058 | 1 693 | 3 386 |
| 7 | 3 521 615 | 65 595 | 104 952 | 209 905 |
| 8 | 218 340 106 | 4 066 901 | 6 507 042 | 13 014 084 |
| 9 | 13 537 086 546 | 252 147 886 | 403 436 617 | 806 873 235 |
| 10 | 839 299 365 868 | 15 633 168 926 | 25 013 070 281 | 50 026 140 563 |

Tab. B.3: Rainbow tables: Znaky z množin a–z, A–Z a 0–9 (celkem 62 znaků).

# C POROVNÁNÍ ALGORITMŮ

| Délka | Symetrický algorimus | RSA | ECDSA | SHA |
|---|---|---|---|---|
| 80 bitů | 2TDEA | RSA 1024 | ECDSA 160 | SHA-1 |
| 112 bitů | 3TDEA | RSA 2048 | ECDSA 224 | SHA-224 |
| 128 bitů | AES-128 | RSA 3072 | ECDSA 256 | SHA-256 |
| 192 bitů | AES-192 | RSA 7680 | ECDSA 384 | SHA-384 |
| 256 bitů | AES-256 | RSA 15360 | ECDSA 512 | SHA-512 |

Tab. C.1: Porovnání síly algoritmů (NIST SP800-57) [47].

| Životnost | Délka | Level |
|---|---|---|
| 2010 | 80 bitů | US Confidential |
| 2030 | 112 bitů | US Confidential |
| | 128 bitů | US Secure |
| | 192 bitů | US Top-Secure |
| Po roce 2030 | 128 bitů | US Confidential |

Tab. C.2: Bezpečnostní životnost (NIST SP800-57, NSA Suite-B) [47].

| Použití | Životnost |
|---|---|
| Digitální podpis | 1 až 3 roky |
| Ověření identity | 1 až 2 roky |
| Symetrické šifrování dat | 2 až 5 let |
| RSA | 1 až 2 roky |
| D-H (Diffie–Hellman) | 1 až 2 roky |

Tab. C.3: Použitelné kryptografické období, životnost klíčů (NIST SP800-57) [48].